

**18th International Conference in Central Europe
on
Computer Graphics, Visualization and Computer Vision**

in co-operation with

EUROGRAPHICS

WSCG 2010

Poster Proceedings

Edited by

Vaclav Skala, University of West Bohemia, Czech Republic

**18th International Conference in Central Europe
on
Computer Graphics, Visualization and Computer Vision**

in co-operation with

EUROGRAPHICS

WSCG 2010

Poster Proceedings

Edited by

Vaclav Skala, University of West Bohemia, Czech Republic

Vaclav Skala – Union Agency

WSCG 2010 - POSTER Proceedings

Editor: Vaclav Skala
c/o University of West Bohemia, Univerzitni 8
CZ 306 14 Plzen
Czech Republic
skala@kiv.zcu.cz

Managing Editor: Vaclav Skala

Published and printed by:
Vaclav Skala – Union Agency
Na Mazinách 9
CZ 322 00 Plzen
Czech Republic

Hardcopy: *ISBN 978-80-86943-86-2*

WSCG 2010

Program Committee members

Adzhiev, V. (U.K.)	Mollá Vayá, R. (Spain)
Balcisoy, S. (Turkey)	Muller, H. (Germany)
Benes, B. (USA)	Murtagh, F. (Ireland)
Bengtsson, E. (Sweden)	Pedrini, H. (Brazil)
Biri, V. (France)	Platis, N. (Greece)
Bittner, J. (Czech Republic)	Puppo, E. (Italy)
Bouatouch, K. (France)	Purgathofer, W. (Austria)
Brodlie, K. (U.K.)	Rojas-Sola, J. (Spain)
Buehler, K. (Austria)	Rosenhahn, B. (Germany)
Csebfalvi, B. (Hungary)	Rudomin, I. (Mexico)
Daniel, M. (France)	Sakas, G. (Germany)
Davis, L. (USA)	Sbert, M. (Spain)
de Geus, K. (Brazil)	Segura, R. (Spain)
Debelov, V. (Russia)	Schumann, H. (Germany)
Ferguson, S. (U.K.)	Sochor, J. (Czech Republic)
Flaquer, J. (Spain)	Stroud, I. (Switzerland)
Gavrilova, M. (Canada)	Teschner, M. (Germany)
Gudukbay, U. (Turkey)	Theoharis, T. (Greece)
Gutierrez, D. (Spain)	Tokuta, A. (USA)
Havran, V. (Czech Republic)	Vergeest, J. (Netherlands)
Chover, M. (Spain)	Weiss, G. (Germany)
Jansen, F. (Netherlands)	Zach, C. (Switzerland)
Kruijff, E. (Austria)	Zara, J. (Czech Republic)
Lee, B. (Korea)	Zemcik, P. (Czech Republic)
Lee, T. (Taiwan)	Zitova, B. (Czech Republic)
Magnor, M. (Germany)	

WSCG 2010

Board of Reviewers

Abas,M. (Malaysia)	Giannini,F. (Italy)
Adzhiev,V. (United Kingdom)	Gonzalez,P. (Spain)
Akleman,E. (United States)	Gudukbay,U. (Turkey)
Aveneau,L. (France)	Guérin,E. (France)
Balcisoy,S. (Turkey)	Gutierrez,D. (Spain)
Battiato,S. (Italy)	Habel,R. (Austria)
Benes,B. (United States)	Hanak,I. (Czech Republic)
Bengtsson,E. (Sweden)	Haro,A. (United States)
Biri,V. (France)	Hasler,N. (Germany)
Bittner,J. (Czech Republic)	Havran,V. (Czech Republic)
Bouatouch,K. (France)	Hernández,B. (Mexico)
Bourdin,J. (France)	Herout,A. (Czech Republic)
Bouville,C. (France)	Horain,P. (France)
Brodlie,K. (United Kingdom)	House,D. (United States)
Bruni,V. (Italy)	Chaudhuri,D. (India)
Buehler,K. (Austria)	Chover,M. (Spain)
Buriol,T. (Brazil)	Jansen,F. (Netherlands)
Camahort,E. (Spain)	Joan-Arinyo,R. (Spain)
CarmenJuan-Lizandra,M. (Spain)	Kohout,J. (Czech Republic)
Casciola,G. (Italy)	Kruijff,E. (Austria)
Csebfalvi,B. (Hungary)	Lanquetin,S. (France)
Daniel,M. (France)	Lee,B. (Korea)
Davis,L. (United States)	Lee,T. (Taiwan)
de Geus,K. (Brazil)	Liu,S. (China)
Debelov,V. (Russia)	Liu,D. (Taiwan)
du Buf,H. (Portugal)	Maciel,A. (Brazil)
Durikovic,R. (Slovakia)	Magnor,M. (Germany)
Erbacher,R. (United States)	Mandl,T. (Germany)
Erleben,K. (Denmark)	Matkovic,K. (Austria)
Feng,J. (China)	Mawussi,K. (France)
Ferguson,S. (United Kingdom)	McMenemy,K. (Ireland)
Ferko,A. (Slovakia)	Michoud,B. (France)
Fernandes,A. (Portugal)	Mokhtari,M. (Canada)
Flaquer,J. (Spain)	Mollá Vayá,R. (Spain)
Galo,M. (Brazil)	Montrucchio,B. (Italy)
Ganovelli,F. (Italy)	Muller,H. (Germany)
Garcia-Alonso,A. (Spain)	Murtagh,F. (Ireland)
Gavrilova,M. (Canada)	Pan,R. (China)

Papaioannou,G. (Greece)
Patane,G. (Italy)
Pedrini,H. (Brazil)
Pina,J. (Spain)
Platis,N. (Greece)
Plemenos,D. (France)
Post,F. (Netherlands)
Pratikakis,I. (Greece)
Puig,A. (Spain)
Puppo,E. (Italy)
Purgathofer,W. (Austria)
Renaud,c. (France)
Richardson,J. (United States)
Ripolles,O. (Spain)
Ritschel,T. (Germany)
Rojas-Sola,J. (Spain)
Rosenhahn,B. (Germany)
Rudomin,I. (Mexico)
Sakas,G. (Germany)
Sanna,A. (Italy)
Sbert,M. (Spain)
Segura,R. (Spain)
Sellent,A. (Germany)
Schneider,B. (United States)
Schumann,H. (Germany)
Sirakov,N. (United States)
Sochor,J. (Czech Republic)

Solis,A. (Mexico)
Sousa,A. (Portugal)
Steinicke,F. (Germany)
Stroud,I. (Switzerland)
Svoboda,T. (Czech Republic)
Teschner,M. (Germany)
Theoharis,T. (Greece)
Theußl,T. (Austria)
Tokuta,A. (United States)
Torrens,F. (Spain)
Tytkowski,K. (Poland)
Vanecek,P. (Czech Republic)
Vasa,L. (Czech Republic)
Veiga,L. (Portugal)
Vergeest,J. (Netherlands)
Vitulano,D. (Italy)
Weiss,G. (Germany)
Wu,S. (Brazil)
Yencharis,L. (United States)
Zach,C. (Switzerland)
Zachmann,G. (Germany)
Zalik,B. (Slovenia)
Zara,J. (Czech Republic)
Zemcik,P. (Czech Republic)
Zhu,Y. (United States)
Zitova,B. (Czech Republic)

POSTER papers

Title	Page
Imanparast,M., Parsaei,M.: A bi-quadratic smooth spline surface generation over irregular meshes	1
Ullrich,T., Schinko,C., Fellner,D.W.: Procedural Modeling in Theory and Practice	5
Kadlec,J., Zemčík,P.: Generation of user interface from characterized code	9
Savadkoohi,B.T., De Amicis,R.: Post- processing of 3D scanning data for custom footwear manufacture	13
Arias-Nicolás,J.P., Calle-Alonso,F.: A novel Content-based Image Retrieval system based on Bayesian Logistic Regression	19
Hans,W., Paulus,D.: Colorimetric Object Classification	23
Behal,D.: Interactive Streamed Media in Virtual Environment	27
Beran,V., Herout,A., Zemčík,P.: On-line Video Synchronization Based on Visual Vocabularies	31
Aracena-Pizarro,D., Mamani-Castro,J.: Museum Guide Through Annotations Using Augmented Reality	35
Juránek,R., Zemčík,P., Herout,A.: Implementing the Local Binary Patterns with SIMD Instructions of CPU	39
Zuzanaák,J., Láník,A., Zemčík,P.: Description of image content by means of graph grammars	43
Ohnishi,Y., Nozaki,K., Baba,K.-I.: Segmentation Technique Based on Object Movement for Speech Production Simulation	49
Shevtsov,M., Soupikov,A.: Efficient acceleration structure layout for 64-bit many-core architectures	53
Özüag,E., Güllü,M.K., Urhan,O., Ertürk,S.: High Resolution 3-D Face Modeling and Model Warping	57
Akhriev,A.: Augmentation of Mean-Shift Method to Facilitate Face Tracking	61
Nishio,K., Takebayashi,Y., Teshima,Y., Kanaya,T., Kobori,K.-I.: Point Cloud Lossless Compression	65
Silveira Jr,L.G., Wu,S.T.: Improving the Responsiveness in Multiplatform Collaborative Environments	71

A bi-quadratic smooth spline surface generation over irregular meshes

Mahdi Imanparast

School of Mathematics, Statistics
and Computer Science, Faculty of science,
University of Tehran,
Tehran, Iran

Majzi28@khayam.ut.ac.ir

Manouchehr Parsaei

School of Mathematics, Statistics
and Computer Science, Faculty of science,
University of Tehran,
Tehran, Iran

Parsaei@khayam.ut.ac.ir

ABSTRACT

A method for generating a smooth spline surface over an irregular mesh is presented. This method generates a smooth spline surface similar to the methods proposed by [Loo94-Pet00, ZZZ+05]. The rules applied to construct the control points in order to achieve the continuity conditions are simple and comprehensible.

Keywords

Spline surfaces, Irregular meshes, Bi-quadratic Bezier patch, Tangent plane continuity, Quad-net.

1. INTRODUCTION

The construction of smooth spline surfaces over control meshes has been a popular topic in computer graphics, geometrical modeling and CAGD. A control mesh consists of a set of vertices, edges and faces.

Irregular meshes differ from regular meshes in the following two ways. Either a vertex has other than four edges emanating from it or a face is defined by other than four edges. To overcome this limitation, different methods have been proposed for the construction of smooth surfaces of irregular topology.

These methods can be roughly categorized into two groups: subdivision surfaces and spline surfaces.

The earliest attempts to overcome the topological limitations of B-spline surfaces were based on the subdivision principle.

Some non-polynomial surface patches used to define B-spline-like surfaces over irregular meshes include the 3 and 5-sided patches defined in [Sab83] and n -sided S-patches in [Loo90]. Hahmann et al. [HB08] presents a piecewise bi-cubic parametric G^1 spline surface interpolating a quad irregular mesh.

A scheme proposed by Peters [Pet93] adjusts irregularities by applying one or more refinement steps. Another scheme by Loop [Loo94] only uses a one refinement step and creates a spline surface. In general this is a composition of patches at most of degree 4. Peters [Pet00] generated a bi-cubic scheme using a Catmull-Clark. Recently, a method has been presented by Zheng et al. [ZZZ+05] in which the Zheng-Ball patches are used to generate a bi-quadratic B-spline-like surface.

In this paper we present a bi-quadratic spline surface, which is a generalization of [Loo94]. The method presented here can be applied in irregular meshes of arbitrary topology. Only one step of subdivision is used. The rules used to generate the control points are simple and comprehensible. It does not go through the complicated computation process needed in [Loo94-Pet00]. That ensures the locality property and has a piecewise polynomial form. Straightforward conditions have been used to ensure smoothness.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

The construction process of this method consists of three steps: the first step carries out a single refinement procedure over the initial mesh, resulting in a new mesh in which the valence of each vertex is four. In the second step a quad-net is constructed corresponding to each vertex of the new mesh. Then each quad-net is split into four sub-quad, and a bi-quadratic Bezier patch is constructed over each sub-quad area.

2. SPLINE SURFACE GENERATION

Constructing the spline surface begins with a user-defined control mesh M^0 . The details of each phase of this method have been presented in the next three sections.

Initial mesh refinement

The first step is to carry out a refinement procedure over initial mesh M^0 . Let F be a face of M^0 consisting of vertices $\{C_0, C_1, \dots, C_{n-1}\}$ and the average of this points is:

$$O = \frac{1}{n} \sum_{i=0}^{n-1} C_i.$$

The point c_i of M^1 corresponding to $\{C_i, F\}$ found by:

$$c_i = \frac{1}{n}O + \frac{n-2}{n}C_i + \frac{1}{2n}C_{i-1} + \frac{1}{2n}C_{i+1}, \quad (1)$$

where all subscripts are taken modulo n .

The faces of M^1 are constructed from vertex, face or edge of M^0 [GRE01] (Figure 1).

Note, that all the vertices of M^1 are 4-valent and every non-4-sided face in new mesh M^1 is surrounded by 4-sided faces. Clearly, if in the initial closed mesh M^0 all vertex valences are already of degree four, this step is not needed and one can use the mesh directly.

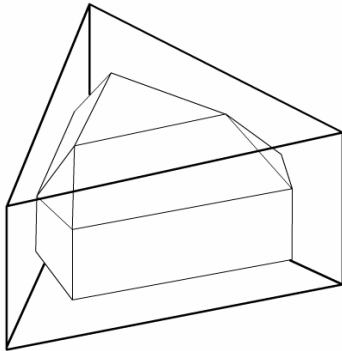


Figure 1. Mesh subdivision. Initial mesh (bold lines). After the subdivision (thin lines).

Generating the Quad-Nets

In the second step, a quad-net is constructed corresponding to each vertex of M^1 .

Consider the vertex V in figure 3, to generate a quad-net on this vertex we will do as follows: The centroid points of four faces surrounding a vertex V are regarded as the corner points of a quad-net ($Q_{00}, Q_{03}, Q_{30}, Q_{33}$).

The points on the boundary of each quad-net are computed such that all quad-net points surrounding a quad-net corner point are coplanar.

The following theorem is the key to constructing quad-net points that satisfy this requirement.

Theorem 2.1: Let $p_0, p_1, \dots, p_{n-1} \in \mathbb{R}^3$ be a set of points in general position. The set of points q_0, q_1, \dots, q_{n-1} found by:

$$q_i = \frac{1}{n} \sum_{j=0}^{n-1} p_j \left(1 + \beta \left(\cos \frac{2\pi(j-i)}{n} + \tan \frac{\pi}{n} \sin \frac{2\pi(j-i)}{n} \right) \right) \quad (2)$$

satisfy:

$$(1 - \cos \frac{2\pi}{n})O + \cos \frac{2\pi}{n}q_i = \frac{1}{2}q_{i-1} + \frac{1}{2}q_{i+1} \quad (3)$$

where:

$$O = \frac{1}{n} \sum_{i=0}^{n-1} p_i$$

and are therefore coplanar.

Proof: take:

$$M_k = \beta \left(\cos \frac{2\pi k}{n} + \tan \frac{\pi}{n} \sin \frac{2\pi k}{n} \right).$$

Expand the right hand side of equation (3) as follows:

$$\begin{aligned} & \frac{1}{2}q_{i-1} + \frac{1}{2}q_{i+1} \\ &= \frac{1}{2n} \sum_{j=0}^{n-1} p_j (1 + M_{j-(i-1)}) + \frac{1}{2n} \sum_{j=0}^{n-1} p_j (1 + M_{j-(i+1)}), \\ &= \frac{1}{2n} \sum_{j=0}^{n-1} p_j (2 + M_{j-(i-1)} + M_{j-(i+1)}), \\ &= \frac{1}{n} \sum_{j=0}^{n-1} p_j \left(1 - \cos \frac{2\pi}{n} \right) + \frac{1}{n} \sum_{j=0}^{n-1} p_j \cos \frac{2\pi}{n} (1 + M_{j-i}), \\ &= (1 - \cos \frac{2\pi}{n})O + \cos \frac{2\pi}{n}q_i. \end{aligned}$$

Note that the well known trigonometric equations of sum of sines and cosines have been utilized in combining $M_{j-(i-1)} + M_{j-(i+1)}$ to get $2 \cos \frac{2\pi}{n} M_{j-i}$.

From relation (3) we can see that q_{i+1} is a linear combination of points O, q_i, q_{i-1} , and by replacement we can find that each q_i ($i \geq 2$) is a linear combination of three points O, q_0, q_1 . Therefore each q_i lies in the plane made by these three points. Hence the collection of q_i 's obtained from the relation (2) are coplanar \square .

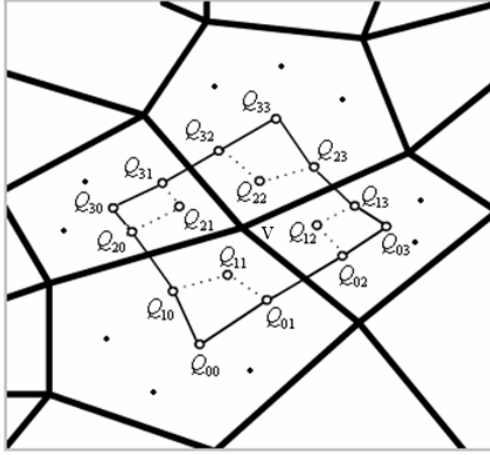


Figure 2. Quad-net.

In theorem 2.1 it can be observed that factor β is a free parameter which can be set arbitrarily.

Interpreting the points p_0, p_1, \dots, p_{n-1} as the vertices of a face blending to mesh M^1 , the point O as Q_{00} and the points q_0, q_1, \dots, q_{n-1} as the quad-net points surrounding Q_{00} . It is immediately clear that all the quad-net points surrounding Q_{00} are coplanar.

Also constraint (3) is satisfied by set:

$$(1 - \cos \frac{2\pi}{n})Q_{00} + \cos \frac{2\pi}{n}Q_{01} = \frac{1}{2}Q_{10} + \frac{1}{2}\hat{Q}_{10}. \quad (4)$$

Where \hat{Q}_{10} is a point of an adjacent quad-net and n is the number of vertices of that face. Similar interpretation are used for the three points Q_{30}, Q_{03} and Q_{33} .

All the boundary quad-net points can be produced easily by applying theorem 2.1 to each one of the four faces surrounding each vertex of M^1 .

The interior point Q_{11} is computed as follows:

$$Q_{11} = Q_{10} + Q_{01} - Q_{00} \quad (5)$$

the other three interior points Q_{12}, Q_{21} and Q_{22} are found by symmetry.

Patch generation

Parametric surface patches are constructed in this step. They interpolate the information generated by quad-nets in the previous step. Each quad-net is accomplished by four bi-quadratic Bezier patches which are constructed as follows: Suppose a quad-net is divided into four bi-quadratic Bezier patches which labeled as shown in figure 3. First we set the corner points:

$$\begin{aligned} a_{00} &= Q_{00}, & a_{04} &= Q_{03}, \\ a_{40} &= Q_{30}, & a_{44} &= Q_{33}. \end{aligned}$$

and:

$$\begin{aligned} b_{11} &= Q_{11}, & b_{13} &= Q_{12}, \\ b_{31} &= Q_{21}, & b_{33} &= Q_{22}. \end{aligned}$$

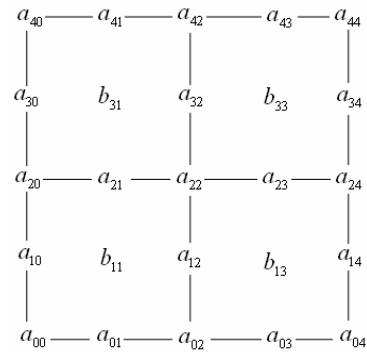


Figure 3. Split of a quad-net into four bi-quadratic Bezier patches.

Formulas for the control points of one of the patches are given here, similar formulas for the other three patches can be found by symmetry.

Internal control points of this patch are:

$$\begin{aligned} a_{12} &= \frac{1}{2}(b_{11} + b_{13}), & a_{21} &= \frac{1}{2}(b_{11} + b_{31}), \\ a_{22} &= \frac{1}{2}(a_{12} + a_{32}). \end{aligned} \quad (6)$$

the other control points are:

$$a_{01} = \frac{1}{2}(b_{11} + b_{31}^{(1)}), \quad a_{10} = \frac{1}{2}(b_{11} + b_{13}^{(2)}). \quad (7)$$

and:

$$a_{02} = \frac{1}{2}(a_{12} + a_{32}^{(1)}), \quad a_{20} = \frac{1}{2}(a_{21} + a_{23}^{(2)}). \quad (8)$$

in which the $b_{31}^{(1)}, b_{13}^{(2)}, a_{32}^{(1)}, a_{23}^{(2)}$ are the points of the two adjacent patches.

It can be observed that these constructions ensure that the triples $\{b_{11}, a_{21}, b_{31}\}, \{b_{11}, a_{12}, b_{13}\}, \{a_{12}, a_{22}, a_{32}\},$

$\{a_{21}, a_{22}, a_{23}\}$ are collinear hence the rectangular patches generated inside each quad-net are C^1 continuous along their internal boundaries.

Two examples of initial meshes and their result surfaces are shown in figure 4.

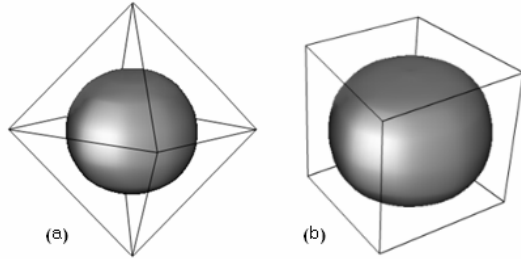


Figure 4. Two models generated by this method.

The same procedure is applied to a boundary quad-net, the only difference is in the control points along the boundary edge.

As it was pointed out in subsection 2.1 if all vertex valences in the initial mesh are of degree four we don't need a refinement step and can immediately create quad-nets, this leads to fewer patches. An example of this special case is presented in figure 4(a).

3. SMOOTHNEES

In this section we establish the smoothness conditions for the resulting surface. As can be seen from the relations of (6), (7) and (8) except in a control point on the corner of the corresponding quad-net each bi-quadratic patches satisfied the C^1 continuity condition over all its control points (shown in figure 5 by the orange points). We now demonstrate that the corner points have tangent plane continuity.

As it was pointed out in subsection 2.2 all the points around the corner of each quad-net found by the theorem 2.1 are coplanar and the internal quad-net points that are computed by the equation (5) lie on the same plane. Finally, the control points of the final patch around this corner point (computed by (7)) are also on the same plane, which includes corner point. This means that, in this corner point we have tangent plane continuity because all the points surrounding it have identical normal vectors.

Considering the symmetric procedure we used to generate the control points in each adjacent patches. It is easily to see that all the patches constructed in this method encode identical tangent planes along the common boundaries.

4. CONCLUSIONS

In this paper, a method is presented to construct a smooth surface over an irregular mesh by means of bi-quadratic Bezier patches. This method can be

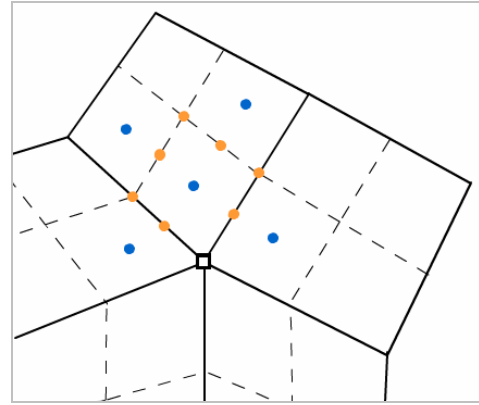


Figure 5. Five quad-nets, common in a corner point. Each bi-quadratic Bezier patch has C^1 continuity in its points (orange points) except in corner point of the quad-net (□) which has the tangent plane continuity.

implemented over both types of open and closed meshes and the result will be a smooth surface.

Following the steps of this method it can be seen that each step has a geometric construction involving weight average (affine combinations) of the points. Therefore, the spline surface is affine invariant.

If all internal vertex valences in the initial mesh are of degree four it should be possible to avoid the subdivision step as an optimization.

5. REFERENCES

- [GRE01] Gohen, E., Riesenfeld, R.F., Elber, G.: Geometric Modeling with Splines: an Introduction. Natick, Mass: Ak Peters, pp. 582-587, 2001.
- [HB08] Hahmann, S., Bonneau, G.P., Caramiaux, B.: Bicubic G^1 interpolation of irregular quad meshes using a 4-split. Lecture Notes in Computer Science 4675, pp. 17-32, 2008.
- [Loo90] Loop, C., DeRose, T.: Generalized B-spline surfaces of arbitrary topology. In: Proceedings of SIGGRAPH, pp. 347-356, 1990.
- [Loo94] Loop, C.: Smooth spline surfaces over irregular meshes. In: Proceedings of SIGGRAPH, pp. 303-310, 1994.
- [Pet93] Peters, J.: Smooth free-form surfaces over irregular meshes generalizing quadratic splines. Computer Aided Geometric Design. 10:347-361, 1993.
- [Pet00] Peters, J.: Patching Catmull-Clark meshes. In: Proceedings of SIGGRAPH, pp. 255-258, 2000.
- [Sab83] Sabin, M.A.: Non-rectangular surfaces suitable for inclusion in a B-spline surface. In: Hagen, T. (ed.) Eurographics, 56-69, 1983.
- [ZZZ+05] Zheng, J.J., Zhang, J.J., Zhou, H.J., Shen, L.G.: Smooth spline surface generation over meshes of irregular topology. The visual computer, 21:858-864, 2005.

Procedural Modeling in Theory and Practice

T. Ullrich, C. Schinko, D. W. Fellner

Institut für ComputerGraphik und WissensVisualisierung, TU Graz, Austria
Fraunhofer Austria Research, Visual Computing Division, Graz, Austria
Fraunhofer Institute for Computer Research and Technical University of Darmstadt, Germany

ABSTRACT

Procedural modeling is a technique to describe 3D objects by a constructive, generative description. In order to tap the full potential of this technique the content creator needs to be familiar with two worlds – procedural modeling techniques and computer graphics on the one hand as well as domain-specific expertise and specialized knowledge on the other hand.

This article presents a JavaScript-based approach to combine both worlds. It describes a modeling tool for generative modeling whose target audience consists of beginners and intermediate learners of procedural modeling techniques.

Our approach will be beneficial in various contexts. JavaScript is a wide-spread, easy-to-use language. With our tool procedural models can be translated from JavaScript to various generative modeling and rendering systems.

Keywords: Computational Geometry and Object Modeling, Computer Graphics Methodology and Techniques, Modeling Languages

1 INTRODUCTION

Within the last few years generative modeling techniques have gained attention. In order to accelerate the modeling process, many researchers enforced research on procedural modeling. All models with well-organized structures and repetitive forms benefit from procedural model descriptions. In these cases generative modeling is superior to conventional approaches.

Its strength lies in a compact description [1], which does not depend on the counter of primitives but on the model's complexity itself. Especially large scale models and scenes – such as plants, cities, and landscapes – can be created efficiently. In this way generative descriptions make complex models manageable as they allow to identify a shape's high-level parameters [7].

A characteristic of generative modeling is its explicit analogy of 3D modeling and programming. This analogy has two negative aspects. First of all, the need to use a programming language is a significant inhibition threshold especially for architects, designers, etc. who are seldom experts in computer science and programming. Secondly, a programming language introduces a new dimension of complexity and further dependencies.

Furthermore, it has never been easy to convert 3D models between various file formats and with generative modeling techniques the situation will probably become worse. If a 3D model does not only contain static geometry but algorithmic descriptions, the

file format also depends on the languages and the interpreter that is able to execute the script.

In this paper we investigate generative modeling approaches concerning these aspects and present a new possibility to create procedural models in a beginner-friendly way. Additionally, we address the file format problem and present a solution to reduce the dependencies to scripting and rendering engines.

2 PROCEDURAL MODELING

In today's procedural modeling systems, grammars are often used as a set of rules to achieve a description. Early systems based on grammars were Lindenmayer systems [17], or L-systems for short. They were successfully applied to model plants. Given a set of string rewriting rules, complex strings are created by applying these rules to simpler strings. Starting with an initial string the predefined set of rules form a new, possibly larger string. The L-systems approach reflects a biological motivation. In order to use L-systems to model geometry an interpretation of the generated strings is necessary. The modeling power of these early geometric interpretations of L-systems was limited to creating fractals and plant-like branching structures. This led to the introduction of parametric L-systems. The idea is to associate numerical parameters with L-system symbols to address continuous phenomena which were not covered satisfactorily by L-systems alone.

CGA Shape

Later on, L-systems and shape grammars were successfully used in procedural modeling of cities [16]. Parish and Müller presented a system that, given a number of image maps as input, generates a street map including geometry for buildings. For that purpose L-systems have been extended to allow the definition of global objectives as well as local constraints. However, the use of procedurally generated textures to represent facades of

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WSCG'2010, February 1 – February 4, 2010
Plzen, Czech Republic.

buildings limits the level of detail in the results. In later work, Müller et al. describe a system [14] to create detailed facades based on the split grammar called CGA shape. A framework called the *CityEngine* provides a modeling environment for CGA shape. It relies on different views to guide an iterative modeling process.

Another modeling approach presented by Lipp et al. [11] following the notation of Müller [13] deals with the aspects of more direct local control of the underlying grammar by introducing visual editing. The idea is to allow modification of elements selected directly in a 3D-view, rather than editing rules in a text based environment. Therefore principles of semantic and geometric selection are combined as well as functionality to store local changes persistently over global modifications.

Model Graphs

Lintermann et al. proposed a modeling method as well as a graphical user interface (GUI) for the creation of natural branching structures [10]. A structure tree represents the modeling process and can be altered using specialized components describing geometry as well as structure. Another type of components can be used for defining global and partial constraints. Components are described procedurally using creation rules which include recursion. The generation of geometric data according to the structure tree is done via a tree traversal where the components generate their geometrical output.

The procedural modeling approach [4] proposed by Ganster et al. describes an integrated framework based on structure trees in a visual language. The infix notation of the language requires the use of variables which are stored on a heap. A graph structure represents the rules used to create an object. Special nodes allow the creation of geometry, the application of operators as well as the usage of control structures. Various attributes can be set for nodes used in a graph. Directed edges between nodes define the order of execution, in contrast to a visual data flow pipeline (VDFP) where data is transported between the different stages.

Hierarchical Description

Finkenzeller presented another approach for detailed building facades [3] called ProcMod. It features a hierarchical description for an entire building. The user provides a coarse outline as well as a basic style of the building including distinguished parts and the system generates a graph representing the building. In the next step, the system traverses the graph and generates geometry for every element of the graph. This results in a generated, detailed scene graph, in which each element can be modified afterwards. The current version has

some limitations: for example, organic structures and inclined walls cannot be modeled.

Scripted Modelers

3D modeling software packages like Autodesk MayaTM provide a variety of tools for the modeling process. In addition to a graphical user interface, a scripting language is supplied to extend its functionality. It enables tasks that cannot be achieved easily using the GUI and speeds up complicated or repetitive tasks.

When using parametric tools in modern CAD software products, geometric validity is a subject. For a given parametric model certain combinations of parameter values may not result in valid shapes. Hoffmann and Kim propose an algorithm [8] that computes valid parameter ranges for geometric elements in a plane, given a set of constraints.

Postfix Expressions

Havemann proposes a stack based language for creating polygonal meshes called Generative Modeling Language (GML). The postfix notation of the language is very similar to that of Adobe's Postscript. It allows the creation of high-level shape operators from low-level shape operators. The GML serves as a platform for a number of applications because it is extensible and comes with an integrated visualization engine.

An extended system presented by Mendez et al. combines semantic scene-graph markups with generative modeling in the context of generating semantic three dimensional models of underground infrastructure [12]. The idea is to connect a geospatial database and a rendering engine in order to create an interactive application. The GML is used for on-the-fly generation of procedural models in combination with a conventional scene graph with semantic markup. An augmented reality view of underground infrastructure like water or gas distribution systems serves as a demo application.

WebGL and O3D

WebGL is a JavaScript binding to OpenGL ES 2.0 which enables rich 3D graphics within browsers on platforms supporting the OpenGL or OpenGL ES graphics standards [9]. A main advantage of this upcoming standard is its plugin-free realization within the browser. The WebGL standard will benefit from recent developments in Web technology like the HTML5 specification or the JavaScript performance increases across all major browsers.

Another "Web"-approach is O3D. This is a combination of a browser plugin and a JavaScript API which enables a web developer to create and display 3D scenes [5]. In order to have more control over the performance of the display routines a plugin is used. The JavaScript part is responsible for the control of the plugin.

Both techniques are still under development so that they can hardly be discussed in detail. However, due to

the fact that JavaScript is used as scripting environment, we will be able to support these techniques as soon as they reach a stable status.

3 LANGUAGE ELEMENTS FOR MODELING

When trying to combine different approaches for procedural modeling, a question arises: Is it possible to achieve a conversion between file formats, respectively languages? The simple answer is: Yes, but only with considerable expenditure. Because of differences in the intended purpose of the languages as well as paradigmatic variations it is a rather difficult task. In order to be able to cover a variety of approaches it would be necessary to implement converters that differ in the source as well as in the target language. Therefore a central solution representing a common ground for the procedural modeling approaches is desirable. This solution enables the user to create procedural models represented by a single language, but allows a variety of output representations to be generated.

Consequently, the motivation for this work is to establish a beginner friendly programming language for procedural modeling that serves as a basis to generate code for different target language. In particular the requirements for such a language can be summarized as follows:

- The language should support a user in typical modeling tasks; i.e. it should provide often needed data structures, algorithms and routines for geometric modeling – such as vectors, matrices, etc.
- As our target group is composed mainly of non-computer scientists and creative coders, the language should be beginner friendly and yet powerful. A user with little experience in programming should be able to read the language and use it in a short period of time [6]. More advanced users should not be limited by the language.
- Languages using error-prone techniques (pointers, memory management, etc. [2]) should be omitted; as Niklaus Wirth stated: “The most important decision in language design concerns what is to be left out.”

Currently high-level programming languages can be classified in scripting- and in system-languages. The main difference – according to John K. Ousterhout [15] – is the style of programming: scripts are designed for gluing programs and algorithms together, whereas system languages are designed for complex algorithms and data structures. As a result scripting-languages are mostly type-less and more dynamic than system-languages. Of course, due to just-in-time compilation and even more advanced techniques this separation is not clear-cut.

In the domain of procedural modeling we favor a scripting language, as we believe that system languages

tend to slow down the creative coding process by programming overhead. Scripting languages tend to be more fault-tolerant and the explanatory power of the source code is promoted. The result of these aspects is presented in the next section.

4 MODELING WITH JAVASCRIPT

The programming language JavaScript meets the requirements mentioned above. It is a structured programming language featuring a rather intuitive syntax, which is easy to read and to understand. As source code is more often read than written, a comprehensible, well-arranged syntax is sensible. JavaScript incorporates features like dynamic typing and first-class functions.

But the most important feature of JavaScript is: it is already in use by non-computer scientists – namely designers and creative coders. JavaScript dialects are used in Adobe Flash (called ActionScript), in the Adobe Creative Suite, in interactive PDF files, in Apple’s Dashboard Widgets, in Microsoft’s Active Scripting technology, in the VRML97, in the Re-Animator framework, etc. Consequently, a lot of documentation and tutorials to introduce the language exist.

However, in order to be used for procedural modeling, JavaScript is missing some functionality, which will be added by libraries.

Data structures and libraries for modeling

In the specification of JavaScript no data types representing vectors and matrices are defined. These types are an essential part of computer graphics. Therefore, the *Euclides* compiler includes a mathematical library to correct this drawback.

Modifications of the Language

While using JavaScript for procedural models it is our aim to be compliant to the standard ECMAScript (ECMA 262). Hence, we try to support this standard and do not add new language constructs and features, which would result in errors when using a standard JavaScript engine. During the development process the compiler’s conformance with the JavaScript standard is tested with JavaScript engines of various web browsers.

5 A GENERATIVE META-MODELER

Our meta-modeler approach differs from other modeling environments in a very important aspect: target independence.

A “normal” generative modeling environment consists of a script interpreter and 3D rendering engine. A generative model (3D data structures with functionality) is interpreted directly to generate geometry, which is then visualized by the rendering engine. In our system a model’s source code is not interpreted but parsed into an intermediate representation. After a validation

process it is translated into the target language. The process of

parsing → validating → translating
offers many advantages.

The validation step involves syntax and consistency checks. These checks are performed to ensure a correct intermediate representation and to provide meaningful error messages as early as possible within the processing pipeline. Sensible error messages are one of the most – if not *the* most – important aspect of a beginner-friendly development environment.

The consistent intermediate representation serves as a basis for backend exporters to different languages. As our compiler has been designed to export and translate JavaScript to other languages, it includes mechanisms to map JavaScript methods and data types to the target language as well as mechanisms to wrap already existing libraries of a rendering engine. The *Euclides* compiler uses annotation techniques to control this mapping and wrapping process. These annotations are placed in JavaScript comments to ensure 100% compliance with the JavaScript standard. In this way low-level, platform dependent functions – such as a method to draw a single triangle – are wrapped platform independently. During the bootstrapping process of a new exporter a few low-level functions need to be wrapped in this way. All other functions, methods, etc built upon these low-level routines are converted and translated automatically.

6 CONCLUSION AND FUTURE WORK

The analysis of existing procedural modeling tools shows similarities and differences. While some approaches are all-purpose modelers, others are specialized on certain subjects.

A common subset of data types and language constructs to describe 3D geometry has been identified. We integrated this common subset in the scripting language JavaScript and developed a corresponding compiler called *Euclides*. It is suited for procedural modeling, has a beginner-friendly syntax and is able to generate and export procedural code for various, different generative modeling or rendering engines.

This meta-modeler concept allows a user to export generative model to other platforms without losing its main feature – the procedural paradigm. The source code does not need to be interpreted or unfolded, it is translated. Therefore it can still be a very compact representation of a complex model.

The target audience of this approach consists of beginners and intermediate learners of procedural modeling techniques and addresses creative designers who are seldom computer scientists. These experts are needed to tap the full potential of generative techniques.

In the future we will support further target platforms and will concentrate on advanced compiler features.

ACKNOWLEDGMENT

The authors gratefully acknowledge the generous support from the European Commission for the integrated project 3D-COFORM (3D Collection FORMation, www.3d-coform.eu) under grant number FP7 ICT 231809, as well as from the Austrian Research Promotion Agency (FFG) for the research project METADESIGNER (Meta-Design Builder: A framework for the definition of enduser interfaces for product mass-customization), grant number 820925/18236.

REFERENCES

- [1] René Berndt, Dieter W. Fellner, and Sven Havemann. Generative 3D Models: a Key to More Information within less Bandwidth at Higher Quality. *Proceeding of the 10th International Conference on 3D Web Technology*, 1:111–121, 2005.
- [2] Matt Bishop and Deb Frincke. Teaching Robust Programming. *IEEE Security and Privacy*, 2:54–57, 2004.
- [3] Dieter Finkenzeller. Detailed Building Facades. *IEEE Computer Graphics and Applications*, 28(3):58–66, 2008.
- [4] Björn Ganster and Reinhard Klein. An Integrated Framework for Procedural Modeling. *Proceedings of Spring Conference on Computer Graphics 2007 (SCCG 2007)*, 23:150–157, 2007.
- [5] Google. O3D API. online: <http://code.google.com/apis/o3d/>, 2009.
- [6] David Gries. What have we not learned about Teaching Programming. *IEEE Computer*, 39:81–82, 2006.
- [7] Sven Havemann and Dieter W. Fellner. Generative Parametric Design of Gothic Window Tracery. *Proceedings of the 5th International Symposium on Virtual Reality, Archeology, and Cultural Heritage*, 1:193–201, 2004.
- [8] Christoph M. Hoffmann and Ku-Jin Kim. Towards valid parametric CAD models. *Computer Aided Design*, 33:81–90, 2001.
- [9] Khronos Group. Khronos Details WebGL Initiative to Bring Hardware-Accelerated 3D Graphics to the Internet. online: <http://www.khronos.org/news/press/releases/khronos-webgl-initiative-hardware-accelerated-3d-graphics-internet/>, 2009.
- [10] Bernd Lintermann and Oliver Deussen. A Modelling Method and User Interface for Creating Plants. *Computer Graphics Forum*, 17(1):73–82, 1998.
- [11] Markus Lipp, Peter Wonka, and Michael Wimmer. Interactive Visual Editing of Grammars for Procedural Architecture. *ACM Transactions on Graphics*, 27(3):1–10, 2008.
- [12] Erick Mendez, Gerhard Schall, Sven Havemann, Dieter W. Fellner, Dieter Schmalstieg, and Sebastian Junghanns. Generating Semantic 3D Models of Underground Infrastructure. *IEEE Computer Graphics and Applications*, 28:48–57, 2008.
- [13] Pascal Müller, Peter Wonka, Simon Haegler, Ulmer Andreas, and Luc Van Gool. Procedural Modeling of Buildings. *Proceedings of 2006 ACM Siggraph*, 25(3):614–623, 2006.
- [14] Pascal Müller, Gang Zeng, Peter Wonka, and Luc Van Gool. Image-based Procedural Modeling of Facades. *ACM Transactions on Graphics*, 28(3):1–9, 2007.
- [15] John K. Ousterhout. Scripting: Higher Level Programming for the 21st Century. *IEEE Computer Magazine*, 31(3):23–30, 1998.
- [16] Yogi Parish and Pascal Mueller. Procedural Modeling of Cities. *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 28:301–308, 2001.
- [17] Przemysław Prusinkiewicz and Aristid Lindenmayer. *The Algorithmic Beauty of Plants*. Springer-Verlag, 1990.

Generation of user interface from characterized code

Jaroslav Kadlec

Faculty of Information Technology
Brno University of Technology
Bozotechnova 2
612 66, Brno, CR
kadlecj@fit.vutbr.cz

Pavel Zemcik

Faculty of Information Technology
Brno University of Technology
Bozotechnova 2
612 66, Brno, CR
zemcik@fit.vutbr.cz

ABSTRACT

Automatic generation of the user interface can simplify development of the computer applications. It can help in the development for various target platforms or in simpler testing and algorithm debugging as the user interface can be created for the actual code and platform taking into account many properties. Process of the automatic generation of the user interface can be supported by the data and code characterization. In this paper, an innovative approach using the data and code characterization is presented. The mechanisms and algorithms describing how the data and code characteristics are loaded, the way how objects are transformed into abstract and specific user interface elements, and the process of finalizing user interface is briefly described. As an example, simple media player is described in every step of the user interface generation process.

Keywords

code characterization, user interface, automatic generation

1. INTRODUCTION

Most of the applications in these days are created as one solution for one or more platforms. The applications portable on a multiple platforms are usually using some intermediate framework, such as Java or .NET Framework [Ell06a]. Many devices with the same or a different platform have very much varying properties and capabilities, so that the user interface created for an average system is not always the best option.

Development of the applications for more platforms simultaneously is a complex task and requires developers to have knowledge of all the required target platforms. Also applications developed for a single platform are composed from heterogeneous information about the algorithms, designs, and user interface which fact can render the design difficult. When the algorithms change, design and user interface code has to change too (to get control over the user interface elements).

For the above reasons, automatic generation of the user interface can simplify application development.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

The user interface elements can be generated to reflect algorithm changes. Cross-platform portability can benefit from the creation of user interface considering capabilities and properties of the executing system.

2. PREVIOUS WORK

Number of the systems exists from 1980's that use various techniques for generating of user interface. A level of automation provided by these systems varies from the programming abstraction (e.g. UIML [Abr02a]) to design tools (e.g. ProcSee [PSe05a]), through the mixed systems requiring partial assistance from user interface designer (TERESA [Pat08a]). Such systems that provide some mechanisms to automatically generate user interface often use simple rule-based approach where every type is matched to the specific user interface element (e.g. UBI [UBI05a]). Some systems rely on the type-based declarative model of the information exchanged through the user interface called Abstract User Interface [Pat03a]. In many cases, a user interface was specified explicitly (e.g. UIML) or inferred from a code [Jel04a]. Some systems include additional information about a high level task or dialogue model (e.g. ConcurTaskTrees [Pat97a] or the task models [Bod94a]). Other systems generate the user interface using constraint satisfaction and optimization (e.g. Supple [Gaj08a]).

3. GENERATING USER INTERFACE

The approach presented here is based on data and code characterization [Kad07a] and it is combining rule-based concept for the interface abstraction from the characterized code and constraint satisfaction and optimization for choosing interface objects. General description of this approach was presented in [Kad06a].

The first step based on the approach is the code characterization. The characterized code is analyzed and the user interface is automatically created from the analysis, considering the properties of the platform, device, user preferences, and the user context. The next step is a rule-based creation of the user interface abstraction using the abstract interface objects. Then, the abstract interface objects are converted to the specific interface objects using the optimization and constraint satisfaction. Finally, the user interface is instantiated. Individual steps of the approach will be demonstrated on a “simple media player” example.

Data and Code Characterization

The first important step in the approach is the characterization of both the data and code. The data and code characterization is a process of annotation of the data and code with a special tags carrying important information for the future user interface generation process.

The characterization can be stored in a separate file or a directly in the source files (see Fig. 1) and compiled and linked to the library or assembly.

```
[DataType(Atomic)]
[DataDomain(Measurement, "Time" )]
[DataContinuity(Continuous)]
[DataTransience(Dynamic)]
[DataImportance(0)]
[CodeName("Time Line")]
[CodeDependence("IsMediafileOpened")]
[ParameterRange(0,0)]
public ulong CurrentPosition { get; set;}
```

Figure 1. Example of the characterized code in C# of a value representing position in the currently played media file.

Loading Code Characteristics

After the code and data characterization is done, the data and code characteristics should be loaded into a characterization tree. The characterization tree reflects well the structure of the data types and their properties and can be used for the creation of the abstract interface objects. A process of the creation of the characterization tree is shown in Fig. 2. First, an instance of the structure holding the characterization data is created (1). Second, attributes are parsed and stored in the structure (2).

```
LoadCharacteristics(tree, dataType) :
1. instance = new Characteristics()
2. ParseCharacteristics(instance, dataType);
3. foreach(variable in dataType)
   a. v = LoadCharacteristics(variable, instance);
   b. instance.Data.Add(v);
4. foreach(method in dataType)
   a. m = LoadCharacteristics(method, instance);
   b. m.Params = ParseParams(method);
   c. instance.Code.Add(m);
5. tree.Insert(instance);
```

Figure 2. Creation of a characterization tree.

Next, every variable is parsed recursively (3). If the variable has already been parsed, a reference to the node is saved. Similarly, for the methods, every method is parsed (4) including its parameters and stored in the tree (5). The resulting characterization tree of the media player is shown in the Fig. 3.

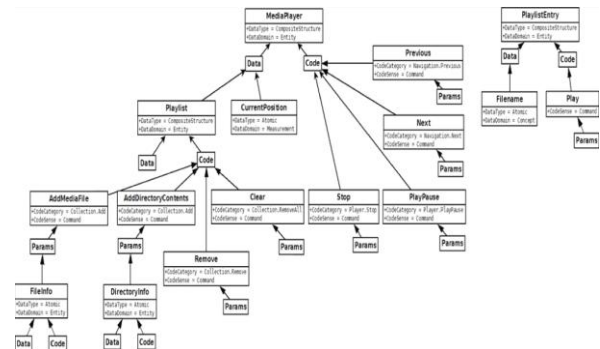


Figure 3. The characterization tree of the simple media player.

Creating Abstract Interface Objects

The Abstract Interface Objects (AIOs) are used to represent a user interface structure. They do not represent specific user interface elements but rather indicate what should be the data or code meaning in the terms of the user interface (the data structure can be e.g. represented as a container with a public data). The process of creation is rule-based with domain limitation and takes into consideration user context and preferences. The process of creation of AIOs is presented in the Fig.4. AIO database is loaded, including all the rules for every AIO. The characterization tree is parsed and for every node in the tree AIO is picked (1) and initialized (2). Initialization for every kind of AIO can be different.

To support a default behavior for commands, a smart template concept [Nic04a] is used (5.a) which groups commands of similar category together. For the methods that require attributes, a dialog AIO is created and filled with AIOs representing each parameter respectively (5.b).

```

Load AIO Database from repository.
CreateAIOs(char, prefs, context)
1. selectedAIO = EvalRuleSet(char, prefs, context);
2. selectedAIO.Initialize(char, prefs, context);
3. char.AIO = selectedAIO;
4. foreach(dch in char.Data)
    a. CreateAIOs(dch, prefs, context);
    b. selectedAIO.Add(dch.AIOs);
5. foreach(cch in char.Code)
    a. if(cch has category && smart template exists)
        i. cch.AIO = GetSmartTemplate(cch.category);
        ii. cch.AIO.Link(cch);
    b. else
        i. cch.AIO = EvalRuleSet(cch, prefs, context);
        ii. if (cch.Params > 0)
            dl原因g = CreateDialogAIO();
        iii. foreach(param in cch.Params)
            1. CreateAIOs(param, prefs, context);
        c. cch.AIO.Add(dlg);

```

Figure 4. Creation of AIO tree.

Fig. 5 demonstrates the resulting AIO tree created from the characterization tree. The main media player object is represented by a container, playlist as a collection, and seek-bar as a time measurement (both are sub-objects of media player class).

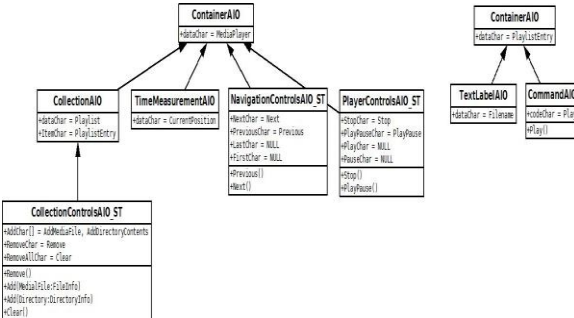


Figure 5. AIOs generated from the characterization tree.

The methods were linked together into three main AIOs thanks to smart template. The Playlist entry was placed separately during the collection initialization because it is used for the internal representation of a collection items and is not explicit part of the media player interface.

Creating Specific Interface Objects

The specific interface objects (CIOs) contain information about the specific user interface element that will be used in the final user interface. The AIO tree with a user interface structure is used to choose the best CIO for every data or code element. The presented process is generally enumeration of all the possible ways of choosing and inserting user interface elements. The best solution with the smallest effort needed for the interaction is chosen. This process is described in Fig. 6. The first step is evaluation of a cost function (1). The cost function evaluates the effort of the user in the interaction with current user interface

objects in his current context and specified device. When the current cost is worse than the best solution found so far, conversion will not continue.

```

ConvertToCIOs(ChTree, AIO, Context, Device)
1. If (CurrentCost(ChTree, Context, Device) >=
    BestCost) return;
2. If (AllCIOsApplied())
    a. BestCost = Cost;
    b. BestCIOs = ChTree.CIOs;
    c. return
3. CIOs = GetCIOs(AIO, Context, Device);
4. foreach(CIO in CIOs)
    a. if(ApplyCIO(CIO, AIO, Device))
        i. subAIOs = GetSubAIOs(AIO);
        ii. SortByImportance(subAIOs);
        iii. foreach(subAIO in subAIOs)
            1. ConvertToCIOs(ChTree, subAIO,
                Context, Device);
5. UndoLastCIO();

```

```

AIOsToCIOs()
1. foreach(SubTree in ChTree)
2. while(true)
    a. ConvertToCIOs(SubTree, SubTree.AIO, Context,
        Device)
    b. if (ConversionComplete(SubTree)) break;
    c. RegroupLowestImportanceContainer(SubTree);

```

Figure 6. Creation of CIO tree.

The second step is checking if all the AIOs were converted to the CIOs and saving solution (2). The third step enumerates all the CIOs available for the concrete AIO (3). Each of these CIOs is applied to the user interface without violating constraints. AIO conversion is repeated for sub AIOs recursively (4). AIOs with a higher importance are always placed first. Finally, the CIO is removed from the user interface because it can be replaced by other CIO in the 4th step of previous recursion.

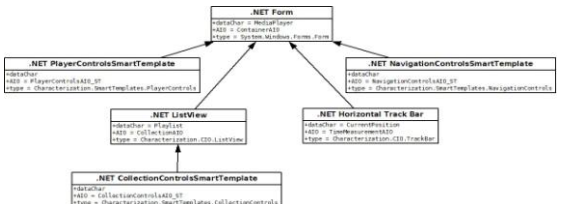


Figure 7. Created CIOs from the AIO tree.

Fig.7. demonstrates the result of the conversion to the CIOs. Main class is represented by the Form (window), containing ListView for the playlist, track bar for the seek-bar and the smart templates for categorized commands.

Instantiating

Instantiation creates instances of CIOs and is responsible for generation and registration of events. The instance of every CIO is created so that instances have the same sub-objects as CIO nodes. Then, the parameters of the CIO are set to the instance. The CIOs representing a data register their dependencies on other objects, events for value

changes of the data and the user interface instance. CIOs representing a code register their dependencies and implementing routines calls, generate events to show asterisks and code to show a dialog for input of the parameters if required.

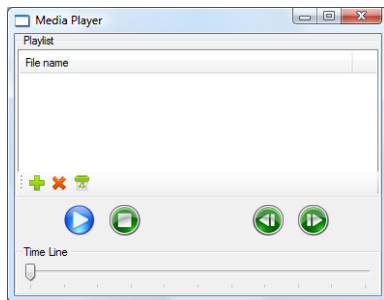


Figure 8. The final user interface.

Fig. 8 shows final user interface generated from CIO tree in Fig. 7. All CIOs were placed in the top-bottom and the left-right order representing highest to lowest importance.

4. CONCLUSION

The presented approach allows for automated creation of the user interfaces from the characterized code. It is based on rule-based creation of the abstract user interface and constraint satisfaction and optimization during creation of specific interface elements. It can be used to create a user interface for various modalities and user contexts. The rule-based approach can quickly reduce the set of the user interface elements for given modality and context while constraint satisfaction and optimization process can create interface with low interaction effort with a device and user capabilities in mind. It can be further extended to consider adaptation to usage. In situations, where the users are getting experienced in time, importance can be changed to reflect the most frequently used user interface elements. In situations, in which the user changes the environment, the current modality can be changed to enable more comfortable interaction.

5. ACKNOWLEDGEMENT

This work was supported by the Security-Oriented Research in Information Technology project sponsored by CEZ MŠMT under Grant No. MSM0021630528, and the Centre of computer graphics sponsored by MŠMT LC under Code No. LC06008.

6. REFERENCES

[Abr02a] Ali, M.F., Pérez-Quñones, M.A., Abrams, M., Shell, E., Building Multi-Platform User Interfaces with UIML. CADUI 2002, France, pp. 255-266, 2002.

[Bod94a] Bodart, F., Hennebert, A.M., Leheureux, J.M., Provot, I., Vanderdonckt, J., A Model-Based Approach to Presentation: A Continuum from Task Analysis to Prototype, in Proc. of FIEW - DSVIS, pp: 25–39, 1994.

[Ell06a] Elliot, S. J2EE and .NET Applications: Creating Value Through Integrated Cross-Platform Management, IDC, 2006.

[Gaj08a] Gajos, K.Z., Wobbrock, J.O., Weld, D.S., Improving the performance of motor-impaired users with automatically-generated, ability-based interfaces, in Proc. of 26th SIGCHI Conf. on HFCS, Italy, pp. 1257-1266, 2008.

[Jel04a] Jelinek, J., Slavik, P., GUI Generation from Annotated Source Code, in Proceedings of Tamodia'04, Prague, Czech Republic, 2004.

[Kad06a] Kadlec, J. Steps In Automated User Interface Generation, in Proc. of SCCG 2006, Bratislava, pp. 25-28, 2006.

[Kad07a] Kadlec, J., Code Characterization for Automatic User Interface Generation, In: Innovations and Advanced Techniques in CISSE, Dordrecht, NL, Springer, s. 255-260, 2007.

[Nic04a] Nichols, J., Myers, B.A., Litwack, K., Improving Automatic User Interface Generation with Smart Templates, In Intelligent User Interfaces, Funchal, Portugal, pp. 286-288, 2004.

[Pat03a] Paternò, F., Santoro, C. A Unified Method for Designing Interactive Systems Adaptable to Mobile and Stationary Platforms, *Interacting with Computers* 15, Elsevier, pp. 349-366, 2003.

[Pat08a] Paterno, F., Santoro, C., Mantyjarvi, J., Mori, G., Sansone, S., Authoring pervasive multimodal user interfaces. *Int. J. Web Eng. Technol.*, Vol. 4, No. 2, pp. 235–261, 2008.

[Pat97a] Paterno, F., Mancini, C., Meniconi, S., ConcurTaskTrees: A Diagrammatic Notation for Specifying Task Models”, in Proc. of the IFIP TC13 Int. Conf. on HCI, Sydney, pp. 362-369, 1997.

[PSe05a] Randem, H.O., Jokstad, H., Linden, T., Kvilesjo, H., Rekvinn, S., Hornæs, A., ProcSee - The Picasso Successor, Enlarged Halden Programme Group Meeting, Lillehammer, Norway, 2005.

[UBI05a] Nylander, S., Bylund, M., Waern, A. The Ubiquitous Interactor - Device Independent Access to Mobile Services, CADUI 2004, Isle of Madeira, pp. 271-282, 2005.

Post- processing of 3D scanning data for custom footwear manufacture

Bitu Ture savadkoohi

¹Department of Information
Engineering and Computer
Science, University of Trento, Via
Sommarive 14 I-38123,
Povo, Trento, Italy

²Fondazione GraphiTech, Via alla
Cascata, 56/c 38123 Povo,
Trento, Italy
bsavadk@dit.unitn.it

Raffaele de Amicis

¹Department of Information
Engineering and Computer
Science, University of Trento, Via
Sommarive 14 I-38123,
Povo, Trento, Italy.

²Fondazione GraphiTech, Via alla
Cascata, 56/c 38123 Povo,
Trento, Italy
raffaele.de.amicis@graphitech.it

ABSTRACT

Footwear fitting has been using manual measurement of a long time, but developments of laser scanners in the last few years have now made automatic determination of footwear feasible. Despite the steady increase in accuracy, most available scanning techniques cause some deficiencies in the point cloud and a set of holes in the triangle meshes. Moreover, data resulting from 3D scanning are given in an arbitrary position and orientation in a 3D space. To apply sophisticated modeling operations on these data sets, substantial post-processing is usually required. In this paper, we described an algorithm for filling holes in triangle mesh. First, the advance front mesh technique is used to generate a new triangular mesh to cover the hole. Then, the Poisson equation is applied to optimize the new mesh; then the models are aligned by "Weighted" Principle Component Analysis.

Keywords

Advance front mesh, Poisson equation, "Weighted" Principle Component Analysis.

1. INTRODUCTION

Due to the rapid development of 3D scanning that can easily and quickly acquire enormous number of surface points from physical parts, reverse engineering (RE) and automatic alignment have become an important steps in the design, manufacturing of new products and shape analyses.

There has been a growing trend among shoe manufacturers to introduce customized shoes to satisfy varying customer comfort needs. The design of new shoes starts with the design of the new shoe last. A shoe last is a wooden or metal model of human foot on which shoes are shaped.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Depending on the both of the complexity of the object and the adopted data acquisition technology some areas of the objects outer surface may never be accessible. This induces some deficiencies in the point cloud and a set of holes in the triangle mesh. Moreover, data resulting from 3D scanning are given in an arbitrary position and orientation in the space. Thus, substantial post-processing is usually required before taking these models to footwear application.

Various techniques have been proposed to fill holes in the mesh. Among non-geometric approaches, authors in [1] used a system of geometric partial differential equation derived from image inpainting algorithms for filling in the holes. Davis et al [2] used volumetric diffusion to fill the gaps. Considering the geometric approaches, Barequet and Sharir [3], find a minimum area triangulation of a 3D polygon with dynamic programming method in order to fill holes. Authors in [4] applied a fairing technique based on solving a non-linear fourth order partial differential equation to fill holes.

Hole process that is implemented here is quite similar to [5]. The main stages of the method is applied in

this paper are: covering the holes with advancing front mesh technique; modifying the triangles in initial patch mesh by estimating desirable normals instead of relocating them directly; solving the Poisson equation according to desirable normal and boundary vertices of the hole to optimize the new mesh. After obtaining complete 3D model, the result data must be generated and aligned before taking this models for shape analysis such as measuring similarity between foot and shoe last data base for evaluating footwear fit.

Principle Component Analysis (PCA), also called Karhunen-Loeve transform, aligns a model by considering its center of mass as the coordinate system origin, and its principle axes as the coordinate axes. The purpose of the PCA applied to a 3D model is to make the resulting shape feature vector independent to translation and rotation as much as possible. In analysis, instead of applying the PCA in a classical way (sets of 3D point-clouds) in order to account different sizes of triangle, Paquet and Rioux [6], established weights associated to center of gravity of triangles and Varanic et al [7], used weighting factors associated to vertices. We used the "Weighted" PCA analysis for alignment of 3D models.

This paper is structured as follows: in section 2 and 3 we introduced terminology, hypothesizes and background while in section 4 filling hole in triangle mesh for building complete model is presented and alignment of 3D models based on the "weighted" PCA is described in section 5. Finally, conclusion and remarks are summarized in Section 6.

2. Terminology and hypothesize

A triangle mesh is defined by a set of oriented triangles joining a set of vertices. Two triangle are adjacent if they share a common edge.

A boundary edge is adjacent to exactly one triangle. A boundary vertex is a vertex used to define a boundary edge. Thus, a closed cycle of boundary edges defines a hole. A given hole is assumed to have no island and all mesh models are oriented and manifold. A boundary triangle is a triangle that own one or two boundary vertices.

1-ring triangles of vertex are all triangles that share one common vertex. 1-ring edges of vertex are all edges that share one common vertex and all vertices on 1-ring edges of a vertex (except itself) are called 1-ring vertices of the vertex.

3. Background

The Poisson equation has been used extensively in computer vision [8]. It arises naturally as a necessary condition in the solution of certain variational problems.

The aim of this method is solving an unknown target mesh with known topology and unknown geometry (vertex coordinate). This equation is able to reconstruct a scalar function from a guidance vector filed and boundary condition.

Consider an unknown scalar function, f , the Poisson equation with Dirichlet boundary condition is given by:

$$\nabla^2 f = \nabla \cdot w \quad \text{over } \Omega, \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega} \quad (1)$$

where w is Guidance Vector Field, $\nabla \cdot w = \frac{\partial w_x}{\partial x} + \frac{\partial w_y}{\partial y} + \frac{\partial w_z}{\partial z}$

is the divergence of $w=(w_x, w_y, w_z)$, f^* provides the desirable values on the boundary $\partial\Omega$, and Laplacian operator is $\nabla^2 = (\frac{\partial^2}{\partial x^2}, \frac{\partial^2}{\partial y^2}, \frac{\partial^2}{\partial z^2})$. Thus it can be defined

as least-squares minimization problem:

$$\min_f \iint_{\Omega} |\Delta f - w| \quad \text{with } f|_{\partial\Omega} = f^*|_{\partial\Omega} \quad (2)$$

A discrete vector field on a triangle mesh is defined to be a piecewise constant vector function whose domain is the set of points on the mesh surface. A constant vector is defined for each triangle, and this vector is coplanar with the triangle. For a discrete vector field w on the mesh, its divergence at vertex v_i can be defined with:

$$(div w)(v_i) = \sum_{T_k \in N(i)} \nabla B_{ik} \cdot w|_{T_k} \quad (3)$$

where $|T_k|$ is the area of triangle T_k , N_i is the 1-ring vertices of v_i and ∇B_{ik} is the gradient vector of B_i within T_k . The discrete gradient of the scalar function f on a discrete mesh is expressed as:

$$\nabla f(v) = \sum_i f_i \nabla \phi_i(v) \quad (4)$$

with $\phi_i(\cdot)$, the piecewise linear basis function valued 1 at vertex v_i and 0 at all other vertices begin and f_i begins the value of f at v_i and it is one of the coordinate of v_i . The discrete Laplacian operator can determine as:

$$\Delta f(v_i) = \sum_{v_j \in N_i} \frac{1}{2} (\cot \alpha_{i,j} + \cot \beta_{i,j}) (f_i - f_j) \quad (5)$$

where $\alpha_{i,j}$ and $\beta_{i,j}$ are the two angles opposite to edge in the two triangles sharing edge $(v_i \text{ and } v_j)$ and N_i is the set of the 1-ring vertexes of vertex v_i , see Figure1. Finally discrete Poisson equation is expressed as follows: $\nabla^2 f \equiv div(\nabla f) = \nabla w$

Discrete Poisson equation with known boundary condition can be defined by a linear system as it results:

$$Ax=b \quad (6)$$

Furthermore, the coefficients matrix A is a symmetric positive definite matrix. So the solution of Poisson equation is reduced to solve the sparse linear system where the coefficients matrix A is determined by Eq.5 and the vector b is determined by Eq.3 and

unknown vector x is the coordinate of all vertices on the patch mesh.

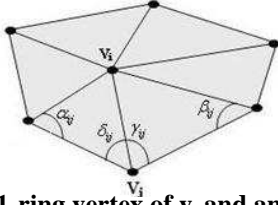


Figure 1. 1-ring vertex of v_i and angles opposite to edge $v_i v_j$

4. Hole filling algorithm

4.1 Identification and hole triangulation

Given the previous hypothesizes, the hole contour can be automatically identified while looking for a closed cycle of boundary edges. The Advance front mesh technique applied over the hole to generate an initial patch mesh is as follow:

First, the angle θ between two adjacent boundary edges at each vertex v_i on the front should be calculated. Next, depending on the angle between e_i and e_{i+1} , the new triangles on the plane should be built, See Figure 2. Then, the distance between new vertex and related boundary vertices is calculated; if distance between them is less than given threshold they should merge. Finally, the front should be updated and the algorithm will be repeated until the hole is patched with new triangle.

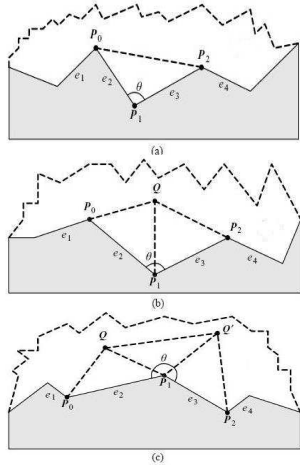


Figure 2. Initial patch mesh generation.

a) $\theta \leq 75^\circ$; b) $75^\circ \leq \theta \leq 135^\circ$; c) $\theta \geq 135^\circ$

4.2 Harmonic normal computation

The most important task of the discrete harmonic function is to map a given disk like surface $S \subset \mathbb{R}^3$ into the plane S^* .

Let V be set of vertices, S_T be a piecewise linear surface that has a boundary, V_B be the set of vertices

lying on the boundary of S_T , V_I be the set of interior vertices of S_T . The goal is to find a suitable (polygonal) domain $S^* \subset \mathbb{R}^2$ and a suitable piecewise linear mapping $f: S_T \rightarrow S^*$. Such a mapping is uniquely determined by the images $f(v) \in \mathbb{R}^2$ of the vertices $v \in V$.

Finite element method based on linear elements is one of the earliest methods for mapping disk-like surface into the plane to approximate a harmonic map. This methods is based on fixing the boundary mapping and minimizing the Dirichlet energy for internal vertices.

$$E_D = \frac{1}{2} \int_{S_T} \|\text{grad}_{S_T} f\|^2 \quad (7)$$

The advantage of current method includes a quadratic minimization and solving linear system of equation. For the triangle $T = \{v_1, v_2, v_3\}$ in the surface S_T . The Dirichlet energy can be expressed as follow:

$$2 \int_T \|\text{grad}_T f\|^2 = \cot \theta_3 \|f(v_1) - f(v_2)\|^2 + \cot \theta_2 \|f(v_1) - f(v_3)\|^2 + \cot \theta_1 \|f(v_2) - f(v_3)\|^2 \quad (8)$$

The normal equations for the minimization problem can be expressed as the linear system of equations

$$\sum_{v_j \in N_i} w_{i,j} (f(v_j) - f(v_i)) = 0 \quad v_i \in V_I \quad (9)$$

where

$$w_{i,j} = \cot \alpha_{i,j} + \cot \beta_{i,j} \quad (10)$$

The angles α_{ij} and β_{ij} are shown in the Figure 1. N_i denotes to 1-ring vertices of vertex v_i . The associated matrix is symmetric and positive, so the linear system is uniquely solvable with sparse and iterative methods such as conjugate gradients methods. Note that the system has to be solved three times, once for the x -, once for the y -, and once for the z -. Now the desirable normal of all vertices in initial patch mesh is obtained.

4.3 Utilizing the Poisson equation to optimize the new mesh

In this section, we imply the Poisson equation according to the desirable normals and the boundary vertices of the hole to approximate the missing geometries more accurately.

Poisson equation requires a discrete guidance field, defined on the triangles of the patch mesh. We applied the local rotation to each triangle of initial patch mesh in order to construct guidance vector field in Poisson equation.

Local rotation can be obtained by rotating original normal of each triangle in patch mesh to new normal of triangle around center of the triangle. After triangle

rotation is performed all triangles on the patch mesh turn to a new direction. So triangles on the patch mesh are not connected anymore and these torn triangles are used to construct a guidance vector filed for the Poisson equation, See Figure 3. Once a discrete guidance filed is given, its divergence at the vertex can be computed.

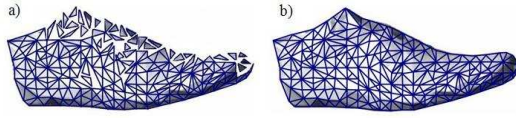


Figure 3. a) Each triangle is locally rotated and triangles becomes disconnected. b) The Poisson equation sticks together new triangles for obtaining smooth triangle mesh

The disconnected triangle are stuck and smooth and accurate patch mesh is reconstructed base on Poisson equation as follow:

First, for each new vertex on their adjacent triangles compute gradient using Eq.4. Next, the divergence of every boundary vertex by using Eq.3 should be calculated. Then, the coefficient matrix A by Eq.5 should be determined and vector b in this equation is determined by using divergence of all boundary vertices. Finally, the new coordinate of all vertices of patch mesh by solving the Poisson equation will be obtained.

5 Alignment of 3D model

The obtained data from 3D scanning are given in arbitrary position and orientation in the space. However for evaluating footwear fit, there is a need for measuring similarity of 3D foot model with shoe last and selecting shoe last from the shoe last data base. Thus, 3D foot model must be properly positioned and aligned before shape analyses such as measuring similarity. We assume the all shoe last in data base are aligned parallel with x-y space from heel to toe. Thus, eigenvectors of all models in shoe last data base are in the same position. see Figure 4.

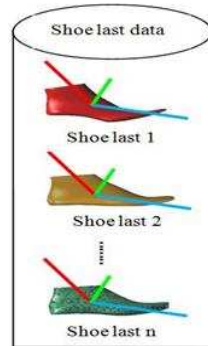


Figure 4. All the models are aligned, parallel with x-y space form heel to toe. Red, green, blue lines are eigenvectors that are in equal positions for all the models

Let $T=\{t_1, \dots, t_n\}$ ($t_i \subset R^3$) be a set of "triangle mesh", $V=\{v_1, \dots, v_n\}$ ($v_i=(x_i, y_i, z_i) \in R^3$) be a set of "vertices" associated to triangle mesh, matrix O_m be the position of eigenvectors of shoe last as column and c_{om} be the "center of gravity of a foot model". The main steps of "weighted" PCA is described the following steps:

Step 1. The translation invariance is accomplished by finding the center of gravity of a model and forming the point set $I=\{v_1 - c_{om}, \dots, v_n - c_{om}\}$.

Step2. The covariance matrix C (type 3*3) can be determined by:

$$C = \begin{bmatrix} cov_{xx} & cov_{xy} & cov_{xz} \\ cov_{yx} & cov_{yy} & cov_{yz} \\ cov_{zx} & cov_{zy} & cov_{zz} \end{bmatrix} \quad (11)$$

$$cov_{xx} = \sum_{i=0}^n \frac{A_i(ct_i x - ctx)(ct_i x - ctx)}{A}$$

Where A be total sum of the areas of all triangles in the mesh, A_i be the area of triangle i within the mesh, let ct_i be "center of gravity of each triangle" and ctx the total sum of "center of gravity" of all of triangles in mesh.

Matrix C is a symmetric real matrix, therefore its eigenvalues are positive real numbers. Then, we sort the eigenvalues in the non-increasing order and find the corresponding eigenvectors. The eigenvectors are scaled to the Euclidean unit length and we form the rotation matrix R, which has the scaled eigenvectors as rows. We apply this matrix to I and we set a new vertex sets called: I'

Step 3. Let matrix N_m be transpose of a matrix R. The alignment is accomplished by constructing a rotation matrix R' through the following formula:

$$R' = O_m \times N_m$$

Step 4. We apply this matrix to I' and calculate new point set .

Figure 5 shows the steps of the current method.

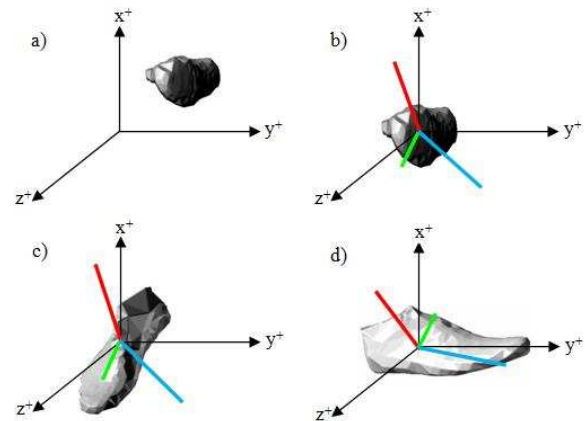


Figure 5. a) Initial Steps of the method. a) Input 3D model. b) Translated center of gravity to the origin. c) Rotated 3D model. d) Alignment of 3D foot with shoe last data base

6 Conclusions

This paper has proposed post-processing steps of 3D scanning data before taking 3D models to the real application.

We proposed a robust and an efficient algorithm for filling holes in triangle mesh. For this purpose, we used the advance front mesh technique to generate initial patch mesh. Then the desirable normals based on the Harmonic function is calculated for modifying initial patch mesh. Finally an accurate and smooth triangle mesh is generated by solving the Poisson equation according to desirable normals and boundary vertices of hole. After obtaining complete 3D models, we applied the "Weighted" Principle Component Analysis technique for alignment of 3D foot with shoe last data base.

After these substantial post processing methods, the 3D foot model is ready for sophisticated modeling operations.

7 REFERENCES

- [1] Verdera J, Caselles V, Bertalmio M, Sapiro G. "Inpainting surface holes", In: International conference on image processing (ICIP); 2003.
- [2] Davis J., Marschner S., Garr M., Levoy M.: "Filling holes in complex surfaces using volumetric diffusion". First International Symposium on 3D Data Processing, Visualization and Transmission Padua, Italy, 2002.
- [3] Barequet G., Sharir M.: "Filling Gaps in the Boundary of a Polyhedron", vol. 12(2):207-229. Computer-Aided Geometric Design, 1995.
- [4] Schneider R, Kobbelt L. "Geometric fairing of irregular meshes for free-form surface design", Computer-Aided Geometric Design 2001.
- [5] Zhao W., Gao S., Lin H.: "A robust hole-filling algorithm for triangular mesh", vol. 23. Computer-Aided Geometric Design, 2007.
- [6] Paquet E., Rioux M.: "Nefertiti: a query by content system for three dimensional model and image databases management", vol. 17:157. Image and Vision Computing, 1999.
- [7] Vranic D., Saup D.: "3D Model Retrieval". Proc.SCCG, 2000.
- [8] Fattal, R., Lischinski, D., and Wearman, M: "Gradient Domain High Dynamic Range Compression", ACM Transactions on Graphics, 2002.

POSTER: A novel Content-based Image Retrieval system based on Bayesian Logistic Regression

J.P. Arias-Nicolás
University of Extremadura, Spain
jparias@unex.es

F. Calle-Alonso
University of Extremadura, Spain
fcalonso@unex.es

ABSTRACT

In this work, a novel content-based image retrieval (CBIR) method is presented. It has been implemented and run on “Qatris IManager” [14], a system belonging to SICUBO S.L. (spin-off from University of Extremadura, Spain). The system offers some innovative visual content search tools for image retrieval from databases. It searches, manages and classifies images using four kinds of features: colour, texture, shape and user description.

In a typical CBIR system, query results are a set of images sorted by feature similarities with respect to the query. However, images with high feature similarities to the query may be very different from the query in terms of semantics. This discrepancy between low-level features and high-level concepts is known as the semantic gap.

The search method presented here, is a novel supervised image retrieval method, based in Bayesian Logistic Regression, which uses the information from the characteristics extracted from the images and from the user’s opinion who sets up the search. The procedure of search and learning is based on a statistical method of aggregation of preferences given by Arias-Nicolás et al. [1] and is useful in problems with both a large number of characteristics and few images.

The method could be specially helpful for those professionals who have to make a decision based in images, such as doctors to determine the diagnosis of patients, meteorologists, traffic police to detect license plate, etc.

Keywords: Computer Vision, CBIR, Logistic Regression, Similarity, Pairwise Comparisons.

1 INTRODUCTION

Research in Content-Based Image Retrieval (CBIR) is today a very active discipline, concentrating on depth issues, such as learning or management access to information content in images. Two fundamental problems remain largely unsolved: how to best learn from users’ query concepts, and how to measure perceptual similarity. A very popular framework in the 1970s was to first annotate the images by text, and then to use text-based Database Management Systems (DBMS) to perform image retrieval. The essential difficulty in this method results from rich content in the images and subjectivity in human perception: the same image content may be perceived differently by other users, [16].

Classification techniques are usually applied in CBIR systems. Image categorization contributes to perform more effective searches. In the repertoire of images under consideration there is a gradual distinction between narrow and broad domains. A broad domain has an unlimited and unpredictable variability in its appearance even for the same semantic meaning [15]. The good performance of classifiers has been proved when the image domain is specific, i.e, it is a narrow domain which has a limited and predictable variability in relevant aspects for the specific purpose, [6, 7].

Actually, there are lots of works directly related to CBIR, and QBIC system is one of the first [8]. It was developed by IBM Corporation and is commercially available. This kind of CBIR systems focus on work with images over a broad domain. Another system is PicHunter [5] which uses Bayesian learning based on a

probabilistic model of user’s behaviour. This system works with features from images and introduce relevance feedback from the users’ opinions. Most recently Liu et al. [12] focused on a powerful feature selection method always addressed to cover the semantic gap; however, these authors do not develop a good method to analyze similarity, but do so for metric distance.

When classification methods are applied to general-purpose image collections the results are not positive, even less if we hope that the performance of the classifier match with the classification developed by non-expert humans. We find some examples in [17, 18].

We thus aim to develop a pairwise comparison method based on the Binary Logistic Regression in order to determine the images that can match one lacking some information. The proposed framework is based on [1], which focused on how to aggregate personal preferences to arrive at an optimal group decision. We are interested in searching for similar images with respect to several features, when we only know the similarity between some pairs of the images. The method for CBIR proposed, combining information of both computational features and user’s knowledge.

2 THE METHOD

2.1 Feature Extraction

Each image is represented by a feature vector of features. We have considered three kinds of features: color, texture and shape features.

- **Color Features.** The color features used in this work are based on HLS model (Hue, Saturation, Luminosity), since the human perception is quite similar to this model. On the other hand, local color features are used in order to achieve information about the spatial distribution [4].
- **Texture Features.** They have been obtained applying two well known methods. The first one works on a global processing of images, it is based on the Gray Level Co-occurrence Matrix proposed by Haralick [10]. The second method is focused on detecting only linear texture primitives. It is based on features obtained from Run Length Matrix proposed by [9].
- **Shape Features.** The images are processed using Active Contours as segmentation method and then some shape features are obtained from these contours. Shape features are based on Hu's moments (first and second moments), centroid (center of gravity), angle of minimum inertia, area, perimeter, ratio of area and perimeter, and major and minor axis of fitted ellipse. The methods to obtain these features are explained in [3].

2.2 Clasification

Once features vectors are obtained, we apply our proposed search method. An efficient images supervised classification method based on bayesian logistic regression has been implemented, which stands out for his high probability of wise move and his facility of incorporating the user's opinion in the learning phase.

The method needs a training stage and a posterior testing one. In the first phase, we require a set of previously classified images to determine the pattern that will be used in the phase of test for classification of new images. One of the important advantages of using the bayesian logistic regression, with regard to others methods, is to obtain a progressive process of the reliability of classifier incorporating the user's opinions and corrections in the successive phases of learning.

The multinomial logistic regression model is a direct generalization of the binary logistic regression for K classes. So, we can classify a new element \mathbf{x} in a class $k \in \{1, 2, \dots, K\}$, for what we can assign to it a K -dimensional vector with values $0 - 1$, $\mathbf{y} = (y_1, y_2, \dots, y_K)^t$ where $y_k = 1$ and the others ones 0. Multinomial logistic regression is a model of conditional probability of the form:

$$p(y_k = 1 | \mathbf{x}, \mathbf{B}) = \frac{e^{(\mathbf{B}_k^t \mathbf{x})}}{\sum_{i=1}^K e^{(\mathbf{B}_i^t \mathbf{x})}}, \quad (1)$$

and standardize by the matrix $\mathbf{B} = (\mathbf{B}_1, \mathbf{B}_2, \dots, \mathbf{B}_K)$. Every column of \mathbf{B} is a vector of param-

eters corresponding to each of the classes: $\mathbf{B}_k = (\beta_{k1}, \beta_{k2}, \dots, \beta_{km})^t$.

The most widely used bayesian approach to the model of logistic regression is to impose a gaussian distribution with mean 0 and variance σ_{kj}^2 , for every parameter β_{kj} ([2, 13]):

$$p(\beta_{kj} | \sigma_{kj}) = \frac{1}{\sqrt{2\pi}\sigma_{kj}} \exp\left(\frac{-\beta_{kj}^2}{2\sigma_{kj}^2}\right). \quad (2)$$

The classification of a new image is based on the vector of conditional probabilities estimated by the model. For this, simply the image is assigned to the class with the highest estimated probability. The maximum likelihood estimation of the parameters \mathbf{B} is equivalent to maximize:

$$L(\mathbf{B}) = l(\mathbf{B} | \mathbf{X}) + \ln p(\mathbf{B}), \quad (3)$$

being

$$l(\mathbf{B} | \mathbf{X}) = -\sum_{i=1}^n \left[\sum_{k=1}^m y_{ik} \mathbf{B}_k^t \mathbf{x}_i - \ln \sum_{k=1}^m e^{(\mathbf{B}_k^t \mathbf{x}_i)} \right] \quad (4)$$

and $p(\mathbf{B})$ is the joint distribution of vector \mathbf{B} .

$L(\mathbf{B})$ is optimized by iteratively maximizing a surrogate function Q , thus (see e.g. [11]):

$$\hat{\mathbf{B}}^{(t+1)} = \arg \max_{\mathbf{B}} Q(\mathbf{B} | \hat{\mathbf{B}}^{(t)}). \quad (5)$$

Once the classifier has been trained, it will be applied to new images whose classification is unknown. The resultant model is applied on a new vector of characteristics to obtain a vector of K probabilities, where K is the number of classes. The k -element of the vector represents the probability of the new image belongs to class k . Therefore, the resultant value to apply the classifier will be the class with major probability of belonging.

2.3 Similarity measure

Having a prior knowledge about the similarity between images, our objective is to find the most similar images to the one given (with respect to the obtained features).

First, we sample r pairs of images from the image database. The idea is to determine a unique discrepancy distance (d_g) for all images, that models the similarity between images.

Then for every pair (a, b) of images, we compute:

- The independent variables, i.e., the distance between their features:

$$\mathbf{x}_{ab} = (d_1(a, b), d_2(a, b), \dots, d_n(a, b)), \quad (6)$$

where d_i is a distance function that models the similarity with respect to the feature i ($i = 1, \dots, n$). Note that, since d_i are distances, these variables are non negative.

- The answer variable or dependent variable, Y_{ab} , is a boolean set of variables, with a value of 0 if the images (a and b) belong to the same class, or 1 if not.

This is the training stage of the system.

We will apply modified Logistic Regression to

$$(x_{j1}, x_{j2}, \dots, x_{jn}, y_j), j = 1, \dots, r.$$

Since our objective is to determine a measurement of discrepancy among images, and as the independent variables are non-negative, the linear predictor will be non-negative. Thus we consider a link function that transforms π into a quantity that takes values in the interval $[0, +\infty)$. Then we consider the link function:

$$g(\pi) = \log \left(\frac{1+\pi}{1-\pi} \right). \quad (7)$$

In this case,

$$\pi = \frac{e^{\mathbf{x}^t \beta} - 1}{e^{\mathbf{x}^t \beta} + 1} \text{ and } 1 - \pi = \frac{2}{e^{\mathbf{x}^t \beta} + 1}. \quad (8)$$

We estimate that parameters $\beta = (\beta_1, \beta_2, \dots, \beta_n)$, which maximize the likelihood function:

$$L(\beta) = \sum_{i=1}^r y_i \log \left(0.5(e^{\mathbf{x}_i^t \beta} - 1) \right) - \log \left(0.5(e^{\mathbf{x}_i^t \beta} + 1) \right),$$

being $\beta \geq 0$.

Unfortunately there is no analytical solution for $\hat{\beta}$ (estimated β), but we may resort to a Newton-Raphson iterative procedure. Each cycle in this procedure provides an updating formula given by:

$$\hat{\beta}^{(k+1)} = \hat{\beta}^{(k)} + (\mathbf{X}^t \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^t \hat{\mathbf{Z}} (\mathbf{Y} - \hat{\mathbf{Y}}), \quad (9)$$

where \mathbf{Y} denotes the vector of response values, \mathbf{X} denotes a matrix with each row by \mathbf{x}_i^t , $\hat{\mathbf{Y}}$ the vector of estimated values at that iteration and \mathbf{W} , $\hat{\mathbf{Z}}$ denotes the diagonal matrix with elements:

$$\hat{z}_i = \frac{e^{\mathbf{x}_i^t \hat{\beta}^{(k)}}}{e^{\mathbf{x}_i^t \hat{\beta}^{(k)}} - 1}, \text{ and}$$

$$\hat{w}_i = y_i \frac{e^{\mathbf{x}_i^t \hat{\beta}^{(k)}}}{(e^{\mathbf{x}_i^t \hat{\beta}^{(k)}} - 1)^2} + \frac{e^{\mathbf{x}_i^t \hat{\beta}^{(k)}}}{(e^{\mathbf{x}_i^t \hat{\beta}^{(k)}} + 1)^2}, \quad (10)$$

respectively. This formula is used until the estimates converge.

We obtain:

$$\hat{\pi}_{ab} = \hat{P}[Y_{ab} = 1] = \frac{e^{\mathbf{x}_{ab}^t \hat{\beta}} - 1}{e^{\mathbf{x}_{ab}^t \hat{\beta}} + 1}, \quad (11)$$

the probability of a and b are different.

Note that for all $a \in \mathcal{A}$, it holds that $d_i(a, a) = 0$ ($i = 1, \dots, n$). Therefore, $\hat{\pi}^{aa} = 0$. Given $a, b \in \mathcal{A}$, we say that a is similar to b if $\hat{\pi}^{ab}$ is close to 0. However if two images a and b are very different with respect to some feature ($d_i(a, b)$ tends to infinite) and $\beta_i > 0$, then the probability that a and b be different, $\hat{\pi}^{ab}$ tends to 1.

We aim to determine the most similar images to the image under study by applying this method. For a new image c , we can compute $\hat{\pi}_{ca_j}$, that can be interpreted as the discrepancy degree between c and a_j . Note that if c and a_j are very similar, the probability of discrepancy is near 0. Then, we must look for the images a_j so that the probability of this one being different from c is near 0. Observe that $\hat{\pi}^{ab}$ near 0 is equivalent to $\mathbf{x}_{ab}^t \hat{\beta}$ near 0. Then, we could consider:

$$d_g(\cdot) = \sum_{i=1}^n \hat{\beta}_i d_i(\cdot), \quad (12)$$

as the measure of discrepancy between images.

2.4 Relevance feedback to searching

The method may improve by providing user's opinion about retrieved images at each step. The β parameter is then updated by applying the Bayesian Logistic Regression method, and by showing new images to the user. A measure is thus obtained for each query image and user. This step corresponds to the learning process in the methodology, see Figure 1.

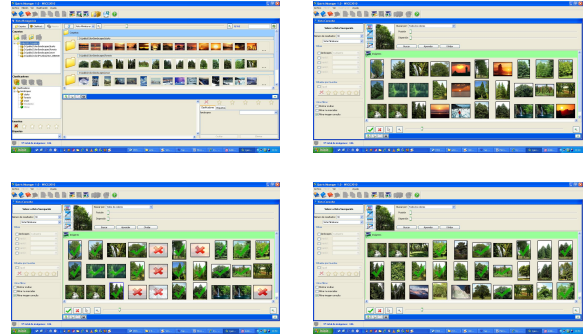


Figure 1: Searching and relevance feedback.

The system displays the most similar images on database to the Query Image (QI). The user can interact with the system, indicating which of the retrieval images (RI) correspond to their search conditions (RI-YES) and which do not (RI-NO). Therefore, the opinion of the user in every query allows to add new pairs by means of a process of feedback, see Table 1.

The system can learn from the answers provided by the user and update vector β with the above mentioned information (feedback). Once optimized, the system offers new similar images. The process repeats itself until the user demonstrates an agreement with the result.

Image	QI	RI-YES	RI-NO
QI	...	$Y = 0$	$Y = 1$
RI-YES	$Y = 0$	$Y = 0$	$Y = 1$
RI-NO	$Y = 1$	$Y = 1$...

Table 1: Learning process (QI: Query Image; RI: Retrieval Image; RI-YES: RI corresponds to the QI; RI-NO: RI does not correspond to the QI).

3 CONCLUSIONS

In this paper we have tested a novel pairwise comparison method to find most similar images and a classification method based on Bayesian logistic regression. The method has sorted query results by similarity, so, we can say that it works like a similarity measure. The technique is based on the Logistic Regression method, and it is useful to get information for Decision Making. In addition the method is integrated in a commercial image database system. It is also particularly useful to solve ranking issues involving a large number of features and few images. Finally, the method is easily applied in practice. By extracting color, texture and shape features from the digital images, acceptable results are obtained for some real CBIR problems. This method proposes an approach to the semantic gap.

ACKNOWLEDGEMENTS

This research was partially supported by project TIN2008-06796-C04-03 from *MEC* and project PDT09A009 from *Junta de Extremadura*, Spain.

REFERENCES

- [1] Arias-Nicolás, J.P., Martín, J., Pérez, C.: A logistic regression-based pairwise comparison method to aggregate preferences. *Group Decision and Negotiation*, **17**(3), 237–247 (2008).
- [2] Berger, J.O.: *Statistical Decision Theory and Bayesian Analysis*. Springer (1985).
- [3] Caro, A., Rodríguez, P.G., Antequera, T., Palacios, R.: Feasible application of shape-based classification. *Pattern Recognition and Image Analysis. LNCS Vol.4477*, pp.588-595 *IbPRIA* (2007).
- [4] Cinque L., Levialdi S., Pellicano A., Olsen K.A.: Color-Based Image Retrieval Using Spatial-Chromatic Histograms. *IEEE International Conference on Multimedia Computing and Systems (ICMCS'99)*, Vol.2, pp. 969-973, (1999).
- [5] Cox, I.J., Miller, M.L., Minka, T., Papathornas, T., Yianilos, P.: The Bayesian Image Retrieval System, PicHunter: Theory, Implementation and Psychophysical experiments. *IEEE Transactions on Image Processing*, **9**, 20–37 (2000).
- [6] Dumais, S., Platt, J., Heckerman, D., Sahami, M.: Inductive learning algorithms and representations for text categorization. *Proc. of 7th International conference on Information and Knowledge Management*. (1998).
- [7] El-Naqa, I., Yang, Y., Galatsanos, N.P., Nishikawa, R.M., Wernick, M.N.: A similarity learning approach to content-based image retrieval: application to digital mammography. *IEEE Transactions on Medical Imaging* **23**(10), 1233–1244 (2004).
- [8] Flickner, M.D., Sawhney, H.S., Niblack, C.W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D., Yanker, P.C.: Query by Image and Video Content: The QBIC System. *IEEE Computer* **28**(9), 23–32 (1995).
- [9] Galloway M. M.: Texture Analysis using Grey Level Run Lengths. *Computer Graphics and Image Processing*, vol. 4, pp. 172-179 (1975).
- [10] Haralick, R.M., Shapiro, L.G.: *Computer and Robot Vision*. Vol.I Addison-Wesley (1993).
- [11] Krishnapuram, B., Carin, L., Figueiredo, M., Hartemink, A.J.: Sparse Multinomial Logistic Regression: Fast Algorithms and Generalization Bounds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**, 6, 957–968 (2005).
- [12] Liu, W., Peng, F., Feng, S., You, J., Chen, Z., Wu, J., Yuan, K., Ye, D.: Semantic Feature Extraction for Brain CT Image Clustering Using Nonnegative Matrix Factorization. *Lecture Notes in Computer Science* **4901**: Medical Biometrics, 41–48 (2008).
- [13] McCullagh, P., Nelder, J.A.: *Generalized Linear Models*, second ed.. Chapman & Hall (1989).
- [14] Qatris Imanager, SICUBO S.L. (www.sicubo.com), Cáceres, Spain.
- [15] Smeulders, A., Worring, M., Santini, S., Gupta, A., Jain, R.: Content-Based Image Retrieval at the End of the Early Years. *IEEE Transactions On Pattern Analysis and Machine Intelligence* **22**(12), 1349–1380 (2000).
- [16] Tversky, A.: Features of similarity. *Psychological Review* **84**(4), 327–352 (1977).
- [17] Vailaya, A., Figueiredo, M., Jain, A., Hong Jiang, Z.: Content-based hierarchical classification of vacation images. *Multimedia Computing and Systems, IEEE International Conference*. Vol. 1, pp. 518-523 (1999).
- [18] Li, J., Wang, J.Z.: ALIP: the Automatic Linguistic Indexing of Pictures system. *Computer Vision and Pattern Recognition. IEEE Computer Society Conference*. Vol 2, pp. 1208-1209 *CVPR* (2005).

Colorimetric Object Classification

Wolfram Hans Dietrich Paulus
Universität Koblenz-Landau
Computational Visualistics
{hans,paulus}@uni-koblenz.de

Abstract

In order to improve object recognition results, usually several image preprocessings are performed. If color images are used, a color normalization is normally applied. Algorithms for color normalization will be compared to a colorimetric approach found in the literature. Recovering colorimetric values instead of a simple *RGB* camera output leads to more reliable color images. To this kind of processed object image a basic object recognition approach using different histogram distances is applied. It will be shown that there is an effect on the results of object recognition rates if we use color calibrated images instead of color normalization methods.

Keywords: Object recognition, color normalization, color calibration

1 INTRODUCTION

The use of color histograms as features is widely used to solve the object recognition task and is described in detail e.g. in [11]. In our contribution, the color of the object is not the main aspect of interest but the statistical distribution of the image itself. We use the *RGB* value and count it in a histogram bin, rather than recording the *color* defined by a colorimetric tristimulus value. We use three histograms per image that are given by each of the *R*, *G*, and *B* color channel in the *RGB* value case as well as in the colorimetric case.

The *RGB* values formed by the camera depends heavily on the image formation process - especially the illumination involved. Mainly for this reason color normalization algorithms are applied to estimate the influence of pose and color of the illumination and eliminate – or at least minimize – their influence to the image appearance. In several situations such color normalizations leads to an improvement of the recognition rates that are presented in section 5. The question arises whether the use of calibrated color values in the histograms lead to another raise of these rates.

The following section gives an overview to some familiar color normalization algorithms. The calibration approach used in the experiments is presented in section 3. Results are discussed in sections 4 and 5.

2 COLOR NORMALIZATION

Using a normalized color one might expect the object features to be better distinguishable. Hence numer-

ous color normalization approaches have been proposed (e. g. [1, 3, 9, 7, 2]).

One very simple method normalizes the color of one pixel (r, g, b) by dividing it by the lightness $(r + g + b)$. The results are called chromaticities.

A very well known assumption is the world to be gray. It was formalized by Buchsbaum [1] and postulates in average the color of a natural scenes sums up to a gray.

A combination of the aforementioned was proposed by Finlayson [3] who iteratively applies the two ideas above in his so-called ‘comprehensive color image normalization’ (CCN) algorithm.

Another idea that uses the gray world assumption is the color normalization by rotation. All color values are considered to be coordinates in a three dimensional color space. All pixel together form a color cloud with a preferred direction. Rotating their principal vector onto the main diagonal, which is the gray axis of the color space, leads to a normalized representation. Several color spaces are used by different authors. One that uses the *RGB* color space is proposed by Paulus [9].

Since color constancy is a nature of human perception, some methods for color normalization try to copy this feature. One well known approach is called ‘Retinex’ and was introduced by Land [7]. A modification used for the experiments is done by Fankle [4].

The Retinex is also capable to normalize the effect of different light sources, while most other algorithms assume only one. One approach that tries to deal with local changes in illumination is called ‘local space average color’ (LSAC) and was introduced by Ebner [2] who assumes that the changes are moderate within the scene. The impact of several color normalization algorithms is shown in Figure 1. Upper left shows the original image from the capturing device, bottom right illustrates the colorimetric calibrated version.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

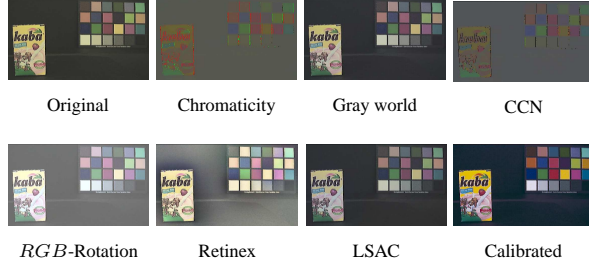


Figure 1: Influence of several color normalization methods used in the experiments to the unprocessed image (top left). Bottom right the result of the calibration method proposed by Lee [8].

3 CALIBRATION

Using a calibration for cameras leads to more reliable results concerning the color of an object. Besides a photometric calibration that determines the brightness transfer function, a radiometric calibration is mandatory for colorimetric information. We use a method that provides a colorimetric calibration as proposed by Lee [8].

3.1 Image formation

A simplified model of image formation considers the light source, the reflectance of the object, and the cameras sensor sensitivity. The brightness transfer function is assumed to be equal to one, making the relation between radiance and the sensors response to be linear. Camera outputs that do not hold this assumption are linearized in a preprocessing step prior to the calibration.

The simplified image formation model is built as

$$f^{(k)} = \int_{\lambda} E(\lambda) \cdot \rho(\lambda) \cdot R_k(\lambda) d\lambda \quad (1)$$

Herein $f^{(k)}$ is used as the sensors response which is the signal at the output of the cameras channel k . Normally, a RGB -triplet value ($K = 3, k \in \{r, g, b\}$) is used. The spectral composition of the light source, denoted by $E(\lambda)$, is multiplied with the spectral reflectance $\rho(\lambda)$ of the object. Hence the cameras incident light is $E\rho$ with the related spectrum. The multiplication with the spectral sensitivity $R_k(\lambda)$ leads to the sensor response which is the integrated value over the spectral range of the sensor. The spectral calculation of the color values in the experiments is limited to the range of 380 nm to 730 nm with respect to the employed measurement device ‘Eye-One Photo’ of Gretag Macbeth. The width of the sampling interval is $\Delta\lambda = 10$ nm which leads to $L = 36$ samples. The discretized version of (1) is

$$f^{(k)} = \sum_{n=1}^L E_{\lambda} \cdot \rho_{\lambda} \cdot R_{k,\lambda} \cdot \Delta\lambda \quad (2)$$

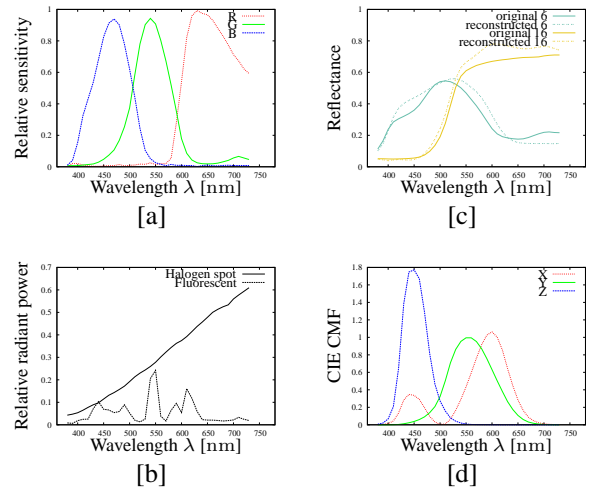


Figure 2: Spectra used for the calibration: Camera sensitivity [a], the light sources used during the image acquisition [b], exemplary two reflectance spectra of the patches 6 and 16 of the ColorChecker[®] with their reconstructed spectra received by the applied calibration [c], as well as the CIE color matching functions used to get the color values XYZ [d].

or rewritten in a matrix notation $\mathbf{f} = \mathbf{P}\mathbf{E}\mathbf{R}$ where \mathbf{f} denotes the $1 \times K$ output vector, \mathbf{P} the $1 \times L$ reflectance vector, \mathbf{E} the $L \times L$ illumination spectrum (a matrix with E_{λ_i} on the diagonal), and \mathbf{R} the $L \times K$ sensor sensitivity matrix. The $\Delta\lambda$ as a constant implicitly is contained in the sensor response.

3.2 Colorimetric calibration

A precondition for calibration as described in Lee [8] is a controlled environment to set the parameters. After calibration, a colorimetric tristimulus can be calculated from RGB values.

The first step is to determine the cameras sensitivity curves. Lee requires for his method an initial estimate $\hat{\mathbf{R}} (L \times K)$ but gives no hint about the necessary quality. For this reason the manufacturers specifications (Figure 2[a]) are used. This first approximation is improved by a 3×3 correction matrix $\tilde{\mathbf{R}}$ that contains an adaption to the prevailing acquisition situation. For this reason an image of the ColorChecker[®] is captured under a known illumination (2[b]). The coefficients of the spectral reflectance of the color patches, (e.g. Figure 2 [c]), are known from the measurements with the photospectrometer, too. The $Q = 24$ RGB -triple that complies with the camera response for each patch is merged into a $Q \times K$ matrix \mathbf{F} . The unknown correction matrix $\tilde{\mathbf{R}}$ is determined by a Moore-Penrose pseudoinverse of $\mathbf{F} = \mathbf{P}\mathbf{E}\tilde{\mathbf{R}}$. The new, adapted camera sensitivity \mathbf{R} is given by multiplication of $\tilde{\mathbf{R}}$ and $\hat{\mathbf{R}}$. One has to keep in mind that this matrix \mathbf{R} has a bias induced by the RGB values of the ColorChecker[®] and the illumination \mathbf{E} .

The next step is to perform a PCA on the spectra of the calibration target. Lee states, that the reconstruction of a spectrum could be done sufficiently correct by three basis vectors that are combined in a $b \times L$ matrix \mathbf{P}_B . The associated weights are called \mathbf{G} ($1 \times b$). Together with the mean spectrum of all patches \mathbf{P}_M , ($1 \times L$) the original spectra can be reconstructed by

$$\mathbf{P} = \mathbf{P}_M + \mathbf{G}\mathbf{P}_B. \quad (3)$$

With this, (2) can be rewritten as

$$\mathbf{G} = (\mathbf{f} - \mathbf{P}_M \mathbf{E} \mathbf{R})(\mathbf{P}_B \mathbf{E} \mathbf{R})^{-1}. \quad (4)$$

This is the central equation for reconstructing a spectrum by calculating the weight \mathbf{G} as a function of the RGB values \mathbf{f} . Putting the weight into (3) leads to a reconstructed spectra.

Since the values are only correct for the patches, in the following several correction steps are performed to get colorimetrically correct values. That is why a $Q \times K$ matrix \mathbf{C} is introduced. For each channel (columns) and patch (rows) the element is determined by the theoretical color values given by $\mathbf{P} \mathbf{E} \mathbf{R}$ divided by the measured RGB values in \mathbf{F} . An additional correction is carried out by multiplying each row of \mathbf{C} with the corresponding ratio of the sum of the reconstructed spectrum to the sum of the measured spectrum. Using $\mathbf{f} = \mathbf{C} \circ \mathbf{F}$ leads now to a colorimetrically correct RGB -value where ‘ \circ ’ denotes the element by element multiplication.

Still, RGB -values that are not part of the 24 ColorChecker[®] patches has to be modified to result in a good spectral estimation. The data has to be interpolated, assuming smooth spectral changes leads to smooth variations in the RGB values.

Hence the spectrum of an unknown triple can be reconstructed by their neighbouring values from known patches. This is done by the Euclidean distance from the \mathbf{f}_r (that has to be reconstructed) to the every element (patch) of \mathbf{F} . This distance is multiplied by a weight factor h to reduce the influence of patches with a larger distance. Lee propose h to be $\ln(10^{-9})$. For the distance to the j^{th} patch $W(j) = e^{\frac{h \cdot D[j]}{\text{range}}}$. These elements are part of the $1 \times Q$ ‘inverse distance weighting vector’ \mathbf{W} . Range means the difference between the minimum and maximum value the camera is capable of, usually 255 using 8 bit. The experiments are performed by using the minimum and maximum value within the image while assuming good-natured content.

When normalizing \mathbf{W} by its sum and multiplying by \mathbf{C} , the result is close to one for those values produced by the patches and zero to those not examined but not exactly one, or zero respectively. This is due to the interpolation performed. For this reason another $L \times L$ correction matrix \mathbf{W}' is applied. Its j^{th} row contains the vector \mathbf{W} described above for the respective patch. To maintain the original correction factor for

the ColorChecker[®] patches a final correction matrix \mathbf{C}' ($L \times K$) is introduced: $\mathbf{C}' = (\mathbf{W}')^{-1} \mathbf{C}$. Using a final correction vector $\mathbf{c} = \mathbf{W} \mathbf{C}'$ to modify the RGB -value \mathbf{f}_r that has to be reconstructed to get the values \mathbf{f} inserted in (4). The resulting weights are used in (3) to reconstruct the spectrum of which two samples are shown in Figure 2 [c].

This spectral estimations are assessed with the color matching functions (Figure 2 [d]) to create the tristimulus values XYZ which we transform to the RGB -color space in our experiments.

4 EXPERIMENTS

The examinations are based on images taken from KOPID¹ that contains 17 objects recorded under three different illuminations and at 12 varying viewpoints. Additionally the camera was moved to five diverse levels of height. Unlike other image databases such as COIL-100² or ALOI³ the KOPID-images contain the ColorChecker[®] what makes it suitable for a colorimetric calibration based on known color samples.

Other than the histogram comparisons in [11] the histograms in the experiment at hand are created for each RGB -channel separately, i.e. we use three one-dimensional histograms. This simplifies the calculation of the Earth Mover’s Distance distance (EMD) [10] which we use for histogram comparison. The information loss compared to the three-dimensional histograms is justifiable by the results which we show below.

The images were subject to one of the color normalizations mentioned above. The quantization of the histograms is done from 8 to 16 bins per channel. As distance measures the sum of squared differences (SSD), histogram intersection (HI), a χ^2 distance, and the EMD are used. The recognition rates are determined as follows: The histogram of the object illuminated by the fluorescent spectrum is compared to all other histograms of the objects illuminated by the halogen spot. If the smallest difference in the histograms is found in an image containing the same object, the recognition task is counted as successful. The recognition rate is calculated by the number of successfully classified objects divided by the total amount of tests performed.

5 RESULTS

In Figure 3 the advance of the EMD (match distance respectively) compared to the other histogram distance measurements is obvious. The rates obtained by the SSD are the least promising. For this reason, the results

¹ <http://www.uni-koblenz.de/kopid>

² <http://www1.cs.columbia.edu/CAVE/software/softlib/coil-100.php>

³ <http://staff.science.uva.nl/~aloi/>

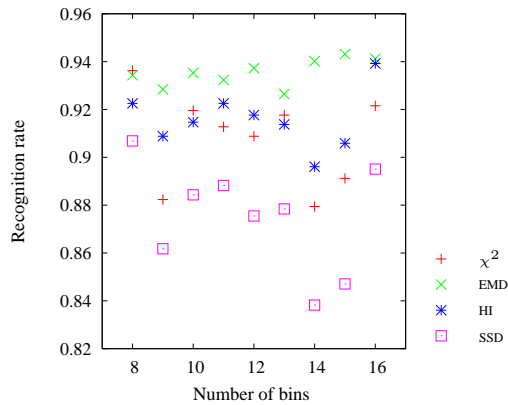


Figure 3: Object recognition rates while using different histogram distance measures. The set used for this case contains the calibrated images only.

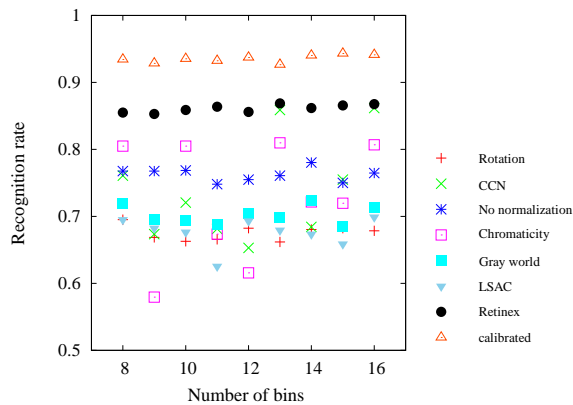


Figure 4: Results using different color normalization algorithms. The blue stars determine the recognition rates where no color normalization is applied. The histogram distance used to obtain this rates is the EMD.

presented in Figure 4 are achieved by the EMD comparison.

The first benchmarks to be reached are the rates received by no normalization (blue stars in Figure 4). These are the recognition rates without performing any color normalization.

It is obvious that some algorithms do not hold the expectation to improve the recognition rate. This is the rotation, gray world, and LSAC. The CCN and chromaticity have marks on both sides of the blue borderline. Using the arithmetic mean they still fail to meliorate the recognition rate of no normalization.

The color normalization performed by the retinex-algorithm displaces the aforementioned by 86,11% correct classifications in average. The Matlab[®] implementation provided by [5] performs a normalization in 8 seconds on a 3GHz computer. That is roughly five times as fast as the colorimetric calibration used.

Nevertheless the calibration applied leads to an additional raise of the recognition rates of around 7% and performs at 93,54% in average on the top of all normalization methods. Compared to the no normalization this is a 17% boost.

6 CONCLUSIONS

To consolidate the results in object recognition rates several extensions could be added. First, the transfer to three-dimensional histograms should be considered. Second, the impact of other color spaces than *RGB* has to be analyzed. Third, the partly heavy variations in some of the normalization methods could be analyzed. One idea here is to increase the number of objects in the database.

The poor performance of some of the algorithms might be a topic for further experiments with other or larger databases or different bin numbers. A larger sample size could also be contemplated for the calibration results - the knowledge of (spectral) image formation given.

Anyhow, finally the results show that a colorimetric calibration outperforms the recognition rates received by commonly used color normalization algorithms. The price of a laborious calibration is well invested into higher recognition rates.

REFERENCES

- [1] G. Buchsbaum. A spatial processor model for object colour perception. *Journal of the Franklin Institute*, 310:1–26, 1980.
- [2] Marc Ebner. Color constancy based on local space average color. *Machine Vision and Applications*, 2008.
- [3] Graham D. Finlayson, Bernt Schiele, and James L. Crowley. Comprehensive colour image normalization. *Lecture Notes in Computer Science*, 1406:475–490, 1998. 1406.
- [4] Jonathan J. Frankle and John J. McCann. Method and apparatus for lightness imaging, 1983. US Patent no. 4,384,336.
- [5] Brian Funt, Florian Ciurea, and John McCann. Retinex in Matlab. In *Proceedings of IS&T/SID Eighth Color Imaging Conference*, pages 112–121, 2000.
- [6] Wolfram Hans, Benjamin Knopp, and Dietrich Paulus. Farbmetrische Objekterkennung. In Gerd Stanke and Michael Pochanke, editors, *15. Workshop Farbbildverarbeitung*, pages 43–51, Berlin, Germany, 9 2009. GfAI.
- [7] E.H. Land and J.J. McCann. Lightness and retinex theory. *Journal of the Optical Society of America*, 61(1):1–11, 1971.
- [8] Raymond L. Lee. Colorimetric calibration of a video digitizing system: algorithm and applications. *Color Research and Applications*, 13(3):180–186, 1988.
- [9] Dietrich Paulus, L. Csink, and Heinrich Niemann. Color cluster rotation. In *Proceedings of the International Conference on Image Processing (ICIP)*, Chicago, 10 1998. IEEE Computer Society, IEEE Computer Society Press.
- [10] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The Earth Mover's Distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.
- [11] Michael J. Swain and Dana H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

Interactive Streamed Media in Virtual Environment

David Běhal

Faculty of mathematics,
physics and informatics

Comenius University
Slovak republic, Bratislava

behal@dejw.sk

ABSTRACT

In our work, we are dealing with problem of displaying video content under given restrictions. This work is oriented on museums and galleries and their exhibitions. The content has to be secured so it cannot be downloaded. With technology of streaming we offer solution for museums and galleries to display their exhibitions in videos. The transferred data is protected by an encryption in order to reduce the possibility of data leakage. The second challenge is to enrich the video with non-standard interaction elements, to make the exhibition more user-friendly and more interactive. Therefore, we propose using the component called “virtual path” for better orientation in museum. This component also visualizes the process of playing the video. Visitors get the opportunity of moving around the museum in chosen order or skipping corridors, which they do not intent to visit. To do that, we suggest adding some additional information about the scene, which will be displayed on demand.

Keywords

Streaming, virtual museum, interactive video, extended interaction

1. INTRODUCTION

In these days more and more museums and also other companies use Web for presenting exhibitions or work. Their presentation consists of certain text, images and also some multimedia elements such as video or audio. First, we have to consider all the possibilities for presentation of the museum.

The simplest way to do this is using a photo gallery like in [EUROPE]. Panoramic photo is more advanced way to present the exhibitions used in [LOUVRE]. However, using pictures has some disadvantages, they are static and they have small

angles of view. On the other hand, the advantage of using photo galleries or panoramas is that they do not take up a lot of space and bandwidth. On the other hand, we can create 3D models of objects and create a virtual walkthrough. This type of presentation is interactive, but it takes a lot of time to make model that will be realistic. 3D models as a presentation of museum is used in [Fer07a].

The last, but not least way of presenting a museum or gallery is to play a video tour around the exhibition. This type of presentation, which was for the first time used in [Cla78a], is very dynamic and video have an audio. Although the visitor can control the walkthrough and jump from one position in video file to another, s/he is not as free as in virtual 3D space.

Our goal is to propose a solution for video in virtual environment, using streaming technology for data transfer and encryption to make the content safe. In addition, we have to extend the standard interaction elements and propose additional interaction.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

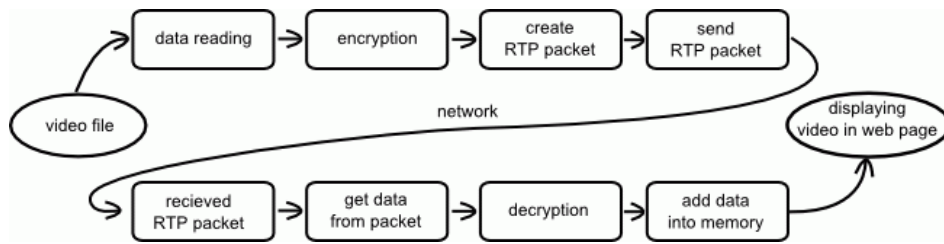


Figure 1: Pipeline from video file to displaying video in web page.

2. BACKGROUND AND RELATED WORK

In this section, we will discuss different ways of displaying a video in web page as well as how to transfer the video data from server to client.

HTML in its last version does not have a tag for adding video into web page. Yet, HTML has tag `<object>` or `<embed>` to embed external video players like Windows Media Player [W3c99a]. Very popular among other web oriented technologies is Flash [FLASH]. The last way is to embed application written in a programming language, e.g. Java.

When we are considering using video file, we have to decide file format (avi, mpg or mov), video format (mpeg-2, mpeg-4 ...) and coding algorithm (DivX, Xvid ...). We will stream the mov file format, as video format we used MPEG-4 specification and the new coding algorithms H.264 [Wie03a].

The video data are saved on server and we have to transfer it for the visitor. Our goal is to forbid downloading the video file, therefore we can stream the data and the user can just see the video and cannot download it. More about video streaming may be found in [Mac02a].

There are many papers that consider streaming technology, so we will mention just a two of them. In [Xu04a], authors propose solution to problem with packet or frame lost while they streamed MPEG-4 video file. The solution to problem of limited storage for video file in one server may be found in [Shi06a].

3. STREAMED MEDIA IN VIRTUAL ENVIRONMENT

In this section we offer our solution for playing streamed media in web page. We will present application named StreamBoat.

3.1. Streaming Data

The first important part is the data transfer. *Figure 1* displays the whole pipeline for transferring data from video file to displayed video. We use *on demand* video streaming [Mac02a], because we have the data already in a file. StreamBoat is a client-server application, where server is responsible for reading data from file and sending the content to the client. Visitor is interacting with client part of application which is a Java Applet. The applet is responsible for playing video, gathering received data and control the server side.

For streaming protocol, we implemented a part of RTP [RTP] for data transfer and RTSP [RTSP] to control the streaming process.

3.2. Data Safety

One of the conditions of this work is to make the content of museum safe from downloading. One of the security measures is streaming technology which erases used data. Although the streaming technology and the RAM memory should secure the content, someone might capture the packets and restore the whole video file. Therefore, we include encryption algorithm into data transfer. The encryption algorithm that we implement is a simple one, but it can be replaced with another algorithm.

3.3. Video in Web Page

To show a video in web page we choose to display the content in Java applet. The package Quick-time Java is controlling and displaying the video content [QTJDOC]. As a consequence, we control the displaying part and data transfer separately. The advantage of this approach is that we can receive the data faster than video plays. However, some data are needed on the client's side before the video starts.

4. INTERACTION

In this section, we will propose and introduce two elements for better interaction. Basically there are three common interaction elements in standard video players, buttons: PLAY, PAUSE, STOP etc., time line to make a linear jump and slider to control the volume. We suggest adding two more elements as virtual path (4.1) and extended interaction (4.2).

4.1. Virtual Path

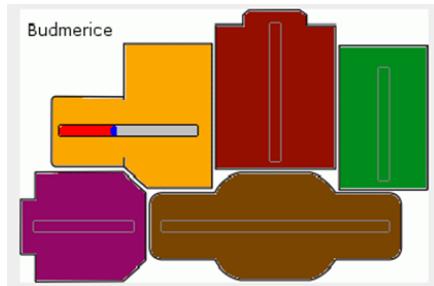


Figure 2: Interactive element called virtual path with ground plan as a background.

As I mentioned before, *Virtual path* (figure 2) is an element for better orientation in the museum and it gives free option in creating visitors own tour. The visitor in the real museum has information about the museum area, like number of rooms or the connections within rooms. We propose to use the ground plan to add this information about the museum as a background of interactive element.

The main concept is to divide the ground plan to rooms or smaller segments like hallways. For every segment there needs to be a separate video file about the corresponding segment. Instead of one timeline we will put separate timelines for every segment into

a corresponding room. These timelines are “connected” with each other in certain places and visitor may choose how s/he will continue the tour. They may be also rotated in every angle. This brings up a *non-linear* walkthrough where the visitor has more choices of direction in the exhibition.

4.2. Additional Information and Extended Interaction

Beside the *virtual path*, we suggest to add some additional information about the objects in video. In work [Son08a] authors propose to add more information about a picture, and they hide it inside the image. We suggest storing the additional information in extra files or database and they will be shown only on visitor’s demand. For this purpose we need an interaction element that will represent them.

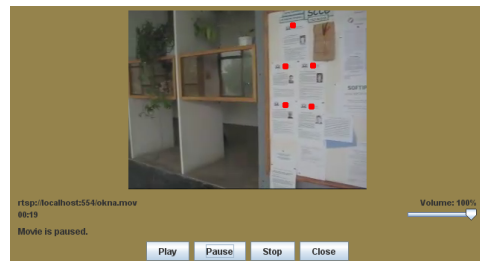


Figure 3: In the video part there are red dots. These dots represent more information about objects.

Let’s imagine an object in our video and we know more information about it. We get the position of the object and also the curve of how it moves. In our solution we have a quadratic B-spline which describes the movement in time. In the actual position of the object we put a dot, clickable object, which will represent some additional data (figure 3).

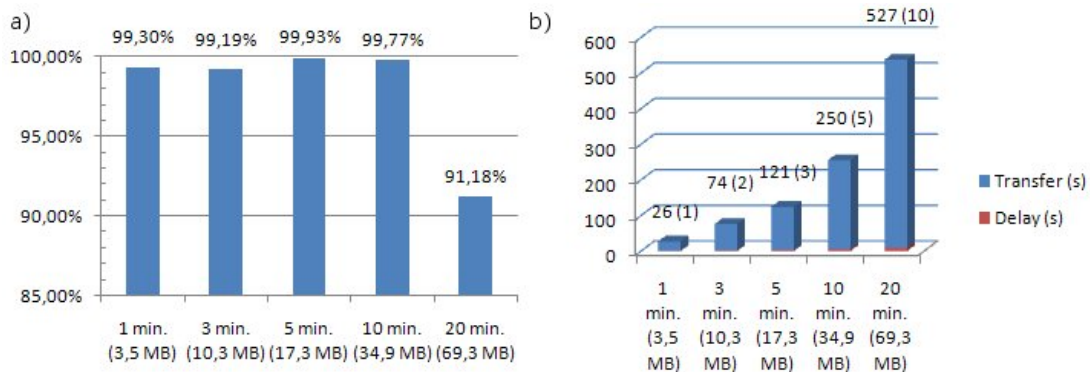


Figure 4a: the data transfer quality test. We streamed 5 videos and we compute the ratio of received data and sent data.

Figure 4b: the blue part - time needed to transfer whole video, the red part - delay before start playing

If the visitor clicks on the dot, s/he will get the additional data. In our solution we have to store just control points of the curve and the start time and end time.

5. RESULTS AND FUTURE WORK

The last part of our work was to make several tests; tests of the quality of data transfer (the criteria were the ratio of sent packages and received packages) and speed test (we have measured the speed of data transfer and the beginning delay). We also had an anonymous inquiry. The server connection for tests was 1,5Mbps down /512Kbps up.

The results of the first specialized test on the quality of data transfer (based on the size of video files) are displayed in *figure 4a*. From the same figure you can see that in all cases the quality was better than 91%. We consider these results as a success. You can find the results of the second test in *figure 4b*.

In the inquiry there were about 30 respondents and mostly (about 65%) they think the application is useful and also like new elements for interaction.

For future work we have some thoughts about another interaction.

6. CONCLUSION

In my work we have connected the art with the technology and proposed the solution for the video in web page problem concerning streaming technology and data encryption for virtual museums and galleries. Our application gives the opportunity to explore the museums from the comfort of your own home and see the exhibition. We extended the standard interactions element with two non-standard. The virtual path gives non-linear movement around the museum and we added additional data to some objects on demand.

7. ACKNOWLEDGMENT

Our project was supported by the Slovak Scientific Grant Agency (VEGA), project No. 1/0763/09.

8. REFERENCES

[Cla78a] Clay, P. et al., Aspen Moviemap Project, MIT, 1978.

- [EUROPE] The Virtual museum of European roots. [Online] [Date: 13. 9. 2009] <http://www.europeanvirtualmuseum.it/>
- [Fer07a] Ferko, A. Virtual Museum Technologies. Virtual University. [Online] 2007. [Date: 10. 10. 2009] <http://www.sccg.sk/~ferko>
- [FLASH] Adobe Systems Incorporated. Flash, [Online] [Date: 11.10.2009] <http://www.adobe.com>
- [JMF] Sun Microsystems, Inc. Java Media Framework API. *Sun Developer Network*. [Online] [Date: 10. 06 2007] java.sun.com/javase/technologies/desktop/media/jmf/.
- [LOUVRE] The museum of Louvre. [Online] [Date: 20. 9. 2009] <http://www.louvre.fr>
- [Mac02a] Mack, S. *Streaming Media Bible*. New York : Hungry Minds, Inc., 2002.
- [QTJDOC] Apple Computer, Inc. QuickTime for Java. [Online] [Date: 15. 12. 2007] developer.apple.com/quicktime/qtjava/.
- [RTP] IETF. RTP: A Transport Protocol for Real-Time Applications. [Online] 1996. [Date: 22. 06. 2009] <http://www.ietf.org/rfc/rfc1889.txt>.
- [RTSP] IETF. Real Time Streaming Protocol. [Online] 1998. [Date: 22. 06. 2009] <http://www.ietf.org/rfc/rfc2326.txt>.
- [Shi06a] Shin, K., et al. MPEG-4 Stream Transmission and Synchronization for Parallel Servers. *IEEE MultiMedia*, 01-03, pp.24-36, 2006.
- [Son08a] Sonnet, et al. Interactive Images using Illustration Watermarks: Techniques, Studies, and Applications. *Proc. Of WSCG2008 Communication papers*, pp. 245-25, 2008.
- [Wie03a] Wiegand, Thomas, et al. Overview of the H.264 / AVC Video Coding Standard. *IEEE Transactions on Circuits and Systems for Video Technology*. 2003.
- [W3c99a] W3C. HTML 4.01 Specification. W3C. [Online] 1999. [Date: 10. 10 2009] <http://www.w3.org/TR/html4/>
- [Xu04a] Xu, H., Diamand, J. and Luthra, A. Client Architecture for MPEG-4 Streaming. *IEEE MultiMedi*, 04-06, pp.16-23, 2004,.

On-line Video Synchronization Based on Visual Vocabularies

Vítězslav Beran, Adam Herout, Pavel Zemčík

Brno University of Technology
Faculty of Information Technology
Department of Computer Graphics and Multimedia
Božetěchova 2, 612 66 Brno, CZ
beranv, herout, zemcik@fit.vutbr.cz

ABSTRACT

This paper presents the procedure for on-line visual-content-based video synchronization. The motivation of our pioneering work is the existence of several off-line video processing systems employed in video classification or summarization applications, but no evidence of on-line solutions for video analysis. In some applications, the video streams go through the broadcast systems that delay the original video and also distort the original signal. The system that would be able to automatically detect such delays, transmission errors, distortions or broadcast failure is highly required. Our solution employs visual vocabularies that allow signing the video frames by bag-of-words. The synchronization procedure is then based on searching for similar frames from different video streams. This paper also overviews the state-of-the-art techniques required for visual vocabulary building and discuss the convenient properties of techniques for real-time on-line systems.

Keywords

On-line video processing, visual vocabulary, video synchronization

1. INTRODUCTION

The main task of the video synchronization systems is to detect and validate the time offset between similar video streams. The motivation of our pioneering work is the existence of several off-line video processing systems employed in video classification or summarization applications, but no evidence of on-line solutions for video analysis. In some applications, the video streams go through the broadcast systems that delay the original video and also distort the original signal. The system that would be able to automatically detect such delays or distortions or detect the transmission errors is highly required.

Our research is focused on visual content, so audio is omitted. Our previous approaches [Ber08] based mostly on global features were poorly robust geometrical transformations that led our research to employ local features and visual vocabularies.

The visual vocabulary used for video content analysis

is introduced in [Siv03]. Motivated by text retrieval techniques, visual ‘words’ are pre-computed using vector quantization and inverted file approach document ranking are used. It results in immediate returning of a ranked list of documents (key frames, shots, etc.) in the manner of search in text documents. The local features are described by SIFT descriptor and quantized using naïve k -means algorithm. When introduced, the visual vocabularies were used in image and object retrieval applications.

The real-time applications such as on-line video synchronization introduce specific demands to the commonly used techniques. The stability and robustness of the local features detection and discriminative power of feature description and quantization could be decreased at the expense of execution time increase. The computational cost demands also derive the size of the visual vocabulary.

The paper firstly introduces the overview of two stages of the video synchronization system. Next sections describe the particular steps of each stage in more detail. Section 3 makes an overview of state-of-the-art image feature extraction and description methods. The searching strategies are described in section 4. The possibilities of visual words weighting when bag-of-words are constructed are discussed in section 5. The sections 6 describes the experiments – used data and explored transformations and discuss the results.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

2. SYSTEM STAGES

The general objective of designed system is on-line monitoring of two or more video streams (query video streams) and detection of possible distortions in space or time domain comparing to the reference video stream.

This work we focused on visual-based video synchronization in time domain. The objectives of the algorithm are:

- real-time image signatures computation,
- real-time image retrieval,
- on-line time shift detection robust to basic video distortions.

The way we utilize the visual vocabularies comprise two stages: off-line vocabulary construction and on-line image retrieval. The steps of both procedures are examined in more detail in following chapters.

The first step of the **visual vocabulary construction** is the extraction of the image features from training dataset. The next section overviews the existed feature detectors in more detail. Then the clustering step divides the features' descriptor space. The clusters' labels are used as visual words. Usually, the iterative k -means clustering algorithm is used. The procedure iteratively re-locates the cluster centers based on mean value of the closest samples to particular cluster. The filtering stage removes (empties) the clusters that do not follow predefined constraints (e.g. cluster size). The various searching strategies are described in section 4 for more details.

The on-line **image retrieval** stage firstly extract features from the image, translate the features' descriptors to visual words using visual vocabulary and compute image signature (bag-of-words).

3. SPARSE FEATURES

In this work, the sparse features mean image key-regions. The key-regions can be constructed on two fundamental image structures: key-points or regions. The following section discusses the most popular approaches. The key-points are expected to be invariant to geometric and illumination changes. Different detectors emphasize different aspects of invariance, resulting in keypoints of varying properties and sampled sizes.

One subset of detectors analyses the local changes in image intensity. The scale-adopted Harris function [Mik04] is sensitive to corner-like structures. The Hessian [Mik05] function, Difference of Gaussian (DoG) [Low04], Laplacian of Gaussian (LoG) [Lin98] detects the blob-like structures. The Harris and Hessian function are extended by characteristic scale detection which is where the LoG function attains a maximum over a scale. To obtain the affine invariance of the detectors, the affine adaptation

process is included that iteratively adapts the region shape by maximizing the intensity gradient isotropy over the elliptical region. Later published results [Bay06] show that the Hessian function can be effectively approximated by block filters. The SURFs are based on effective platform based on computation of Haar-wavelets on integral images. The authors also introduced new descriptors utilizing the same platform.

In this work, the speed factor is the most crucial. The SURF detector offers convenient balance between speed, robustness, precision and discriminative power. We used the SURF detector and descriptor in extended version that gives 128-dimensional feature descriptors.

4. VISUAL VOCABULARY

The idea of visual vocabulary, firstly used in "Video Google" [Siv03], brings the techniques from natural language processing and information retrieval area. The document (image) is represented as an unordered collection of words (bag-of-words model). In computer vision, the (visual) words might be obtained from the feature vectors by a quantization process. The objective is to use vector quantization to descriptors to translate them into clusters' labels which represents the visual words.

Visual vocabulary is created during the training stage. A part of the data (training data) is used to find cluster centers in the descriptor space. When the size of the resulting vocabulary is small ($k < 10^5$), the naïve k -means algorithm can be used. The time complexity of the k -means algorithm is $O(kN)$, where N is the number of training feature vectors. Some applications (e.g. for object retrieval) need more discriminative vocabulary. One possible way how to reduce the time complexity is using Hierarchical k -means [Nis06]. Instead of solving one clustering with a large number of cluster centers, a tree organized hierarchy of smaller clustering problems is solved. This reduces the time complexity to $O(N \log k)$. The problem with HKM is that it optimizes the problem only locally, per tree branch. Other approach reducing the time complexity is replacing the nearest neighbor search of k -means by forest of kd-trees. The authors [Phi07] called this approach as Aproximate k -means.

This work utilizes the idea from AKM with one kd -tree for space search. A random selection of images from training datasets serves to generate feature descriptions. Because the discriminative power of the visual vocabulary video synchronization system does not need to be superior, we built the vocabulary of sizes 1k of visual words.

5. IMAGE SIGNATURE

In our framework, the *image signature* is a collection of weighted words – vector of words’ frequencies. It degrades to a set-of-words when the weights represent only the word’s presence. Otherwise, it is a bag-of-words. In recent works, several weighting schemes were introduced and evaluated.

In the pioneering work [Siv03], standard weighting used in text retrieval is employed that is known as ‘term frequency – inverse document frequency’ – *tf-idf*. The *term frequency* reflects the entropy of a word with respect to each document unlike *inverse document frequency* down-weights words that appear often in the database. The resulting weight is then:

$$\begin{aligned} tf-idf(w) &= tf(w) * idf(w) \\ &= \frac{|d(w)|}{|d|} * \log\left(\frac{|D|}{|D(w)|}\right) \end{aligned} \quad (1)$$

where d is a document (set of words), $|d|$ is a number of words in d and $|d(w)|$ is the number of occurrences of word w in d , D is a dataset of all documents and $D(w)$ is a set of documents containing the word w .

At the retrieval stage, documents are ranked by their similarity. One of the mostly used similarity metric in text retrieval is normalized scalar product (cosine of angle) between the query vector \mathbf{q} and all document vectors \mathbf{d} in the database.

$$sim(\mathbf{q}, \mathbf{d}) = \frac{\mathbf{q} \cdot \mathbf{d}}{\|\mathbf{q}\| \|\mathbf{d}\|} \quad (2)$$

where \cdot is a dot product and $\|\cdot\|$ is the vector magnitude.

For the experiments in this work, the standard *tf-idf* weighting scheme is used when visual words are weighted in bag-of-words.

6. TIME SHIFT DETECTION

The system contains a frame buffer for each video stream. Local features are extracted and described for each incoming frame (query and reference streams) and used to create frame signature. The signatures are stored to the buffer for each video stream and used to compare frames. The similarities between each new frame from query video and all reference frames in the reference buffer are computed.

The similarities between query frame and the frames from buffer of reference frames in time t can be expressed as a function:

$$s(t, i) = sim(\mathbf{q}_t, \mathbf{d}_{t,i}) \quad (3)$$

where \mathbf{q}_t is a signature of the query video frame and $\mathbf{d}_{t,i}$ is the signature of the i -th frame in the reference buffer at time t . Using the function in equation (5), the probability of a time offset i can be expressed as the weighted sum of previous similarities:

$$O(t, i) = w(t) \otimes s(t, i) \quad (4)$$

where w is the weighting function. Two weighting functions were used in experiments – averaging (Eq. 7) and exponential function (Eq. 8):

$$w(t) = \frac{1}{N} \quad (5)$$

$$w(t) = 1 - \log_N t \quad (6)$$

where N is the window (buffer) size. The most probable time offset is then taken as the detection result:

$$offset(t) = \arg \max_i O(t, i) \quad (7)$$

The voting strategy is employed to sum up the most probable time offsets over the time which stabilizes the detection results.

Along with the detected time shift, the certainty level of the detection is computed. The certainty level is computed from the two bins from the voting that have the most votes. The precise function computing the certainty level is:

$$c_level(t) = 1 - \frac{|votes(t)|_2}{|votes(t)|_1} \quad (8)$$

where $votes(t)$ is the ranked list of voting bins in time t and $|\dots|_r$ is the amount of votes in the r -th bin.

7. RESULTS

We use ‘‘TRECVID 2009 data BBC subset’’ to evaluate our system. The subset contains 77 video files. The TRECVID video data contains a variety of challenging disruption itself, especially for time video synchronization task. The TRECVID videos begin with long (cca 40-60 seconds) shots with color initialization stripes where very few local features are detectable. Further, many of the shots are static or almost static, so the frame comparison does not favor any frame and the detection fails; resp. does not vote for any particular time shift.

The query video streams were created artificially. In this work, we focused on fundamental types of video distortions:

- a uniform scale transformation,
- partial occlusion of the visual content by static banner and
- white noise.

All experiments used the visually vocabulary trained on different dataset than the testing one. For training purposes, the Kentucky dataset [Nis06], containing 10200 images, was used. The dataset contains 4-tuples of object images taken from difference viewpoint. Each experiment run was done on all 77 video files and certainty level from all files was aligned and averaged. The alignment is necessary because the length of the initialization shot with color

stripes at the beginning of each video differs. The video frame size is 352x288 pixels.

The system sensitivity to the different time shift was also explored. The reference video was delayed by 1 and 10 seconds. The tests with resized query video to 90% of the original size revealed that in both cases, when query video is delayed by 25 frames and 250 frames, the exponential weighting function outperforms the averaging function. The robustness to partial occlusion of the video frames was tested using overlaid banner. The graph on Figure 6 shows that the precision of the detector is not affected by partial occlusion; at least up to 25 frames delay. The robustness to the noise was tested with the noise level in between 0%-30%. The Figure 7 shows how the detector precision decreases with increasing level of noise. The experiments with blurring have proven the expected properties about the used feature extractor. The SURF features are sensitive to blobs so their repeatability decreases very slowly with increasing amount of blur.

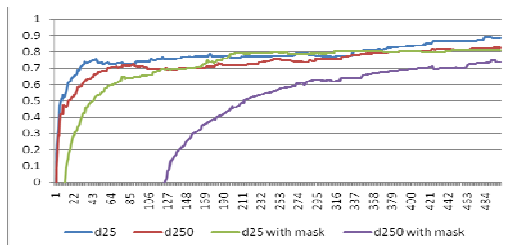


Figure 6. Partial occlusion by a banner.

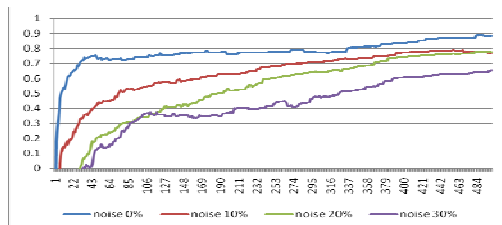


Figure 7. Added noise of different levels.

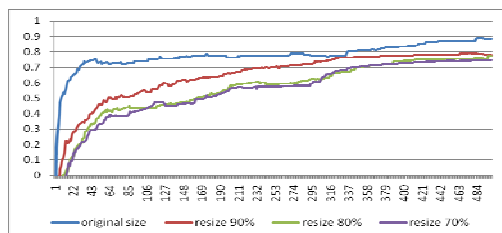


Figure 9. Uniform rescale transformation.

The computational costs of the algorithm depend on amount of extracted features. Data used in evaluation contain cca 200 features per frame. The algorithm was able to process 13 frames per second on the desktop PC Intel Core Duo 2.4GHz, 3.5GB.

8. CONCLUSION

The objective of the presented work was to design and evaluate the real-time on-line system for visual-

content-based video analysis. The solution is based on the visual vocabulary and bag-of-words. Several types of video distortions were addressed. The results of evaluated experiments show that the proposed solution is usable for the on-line real-time video analysis and video content comparison. The solution can be used as the basement for the other methods; e.g. dealing with geometrical transformation between video streams. The knowledge of the geometrical transformation might be useful when non-uniform geometrical distortion is expected and the automatic video quality control task analyzing noise ratio, contrast changes, color bias, etc. is required.

9. ACKNOWLEDGEMENTS

This work has been supported by the Ministry of Education, Youth and Sports of the Czech Republic under the research program LC-06008 (Center for Computer Graphics), by the research project "Security-Oriented Research in Informational Technology" CEZMŠMT, MSM0021630528 and EU IST FP6 projects "AMIDA" EU-6FP-IST, IST-033812-AMIDA.

10. REFERENCES

- [Bay06] Bay, H., Tuytelaars, T., Van Gool, L. *SURF: Speeded Up Robust Features*, Proceedings of the ninth European Conference on Computer Vision, 2006.
- [Ber08] Beran, V., Herout, A., Hradiš, M., Řezníček, I., Zemčík, P. *Video Summarization at Brno University of Technology*. In: ACM Multimedia 2008, New York: ACM, 2008.
- [Lin98] Lindeberg, T. Feature detection with automatic scale selection. *Int. J. of Computer Vision*, 30:79-116, 1998.
- [Low04] Lowe, D. Distinctive image features from scale-invariant keypoints. *Int. Journal on Computer Vision*, 60(2):91-110, 2004.
- [Mik05] Mikolajczyk, K., et. al. A comparison of affine region detectors. *Int. Journal on Computer Vision*, 65(1/2):43-72, 2005.
- [Mik04] Mikolajczyk, K., Schmid, C. Scale and affine invariant interest point detectors. *Int. Journal of Computer Vision*, 60:63-86, 2004.
- [Nis06] Nister, D., Stewenius, H. *Scalable recognition with a vocabulary tree*. In Proc. CVPR, 2006.
- [Phi07] Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A. *Object retrieval with large vocabularies and fast spatial matching*. In Proc. CVPR, 2007.
- [Siv03] Sivic, J., Zisserman, A. *Video Google: A text retrieval approach to object matching in videos*. In Proc. ICCV, 2003.

Museum Guide Through Annotations Using Augmented Reality

Diego Aracena-Pizarro
Universidad de Tarapacá
Computer and Informatics Engineering Area
CHILE, 1010069, ARICA
daracena@uta.cl

Jovanna Mamani-Castro
Universidad de Tarapacá
Computer and Informatics Engineering Area
CHILE, 1010069, ARICA
jovanna.mc@gmail.com

ABSTRACT

The identification of objects displayed in existing Archaeological Museums, requires the observer to recognize the object or the associated concepts in the thematic environment. Usually, the visitor has a written guide or an expert that accompanies him on his visit. This paper presents an automated guide through annotations in Augmented Reality (AR) unmarked fiducial or free marker, by the recognition of objects shown in the showcase of the museum, showing 2D annotations on the objects. For that purpose, using Head Mounted Displays (HMD) or mobile PDA, the observer can see these annotations. The recognition is based specifically on mathematical relationships that gives us the principal component analysis (PCA), which evolves to create vectors that show more clearly the relationships in the image, which are known as principal component vectors (PCV). These concepts are used to identify archaeological objects and to create a symbolic image database for 3D archaeological objects existing in museum's showcases, which allows to conform the museum's guide system in appropriate time.

The work helps in shaping an appropriate proposal of recognition to create the symbolic image database of objects, the proper record of annotations on objects, when the mouse click selects it. This record is achieved by robust calculation, and monitoring based on homographies to ensure the insertion of the adequate annotation.

Keywords

Principal Component Vector (PCV), Wrapping, Homography, Augmented Reality.

1. INTRODUCTION

Augmented Reality (AR) is seen as a variation on the concept of Virtual Environment (VE), which allows the user to see the real world around him combined with superimposed virtual objects or compounded with it.

AR is a specific example of what Brooks, F. [Bro96] called Intelligence Amplification, using the computer to make easier tasks performed by humans. Hence the diversity of application fields, among which is, for example, Maintenance and Repair by Billingham, J. [Bil06], where AR applications simplify training and understanding for assembly, maintenance of complex industrial machinery, where annotation are shown, which can be used to associate them to real objects with information about themselves. Els Vilars [Vil06] in Arbeca, describes an Augmented Reality System developed at Universidad de Lleida, Spain, which using patterns marks placed on the environment, allow visits to the archaeological ruins.

Technological advances and tracking techniques experienced in recent years allow us to explore outdoor environments with mobile systems. These

systems open new possibilities in navigation and the use of geographic information.

Simon, G. et al. [Sim00], performs tracking by planar homography, using the calculation of successive homographies with RANSAC (robust computing homography). The robust estimate of the homography is also used in the work in Malik, S. et al. [Mal02], tracking with passive mark a familiar pattern.

Yuan, M. et al. [Yua06] presents a scheme free of marks to achieve tracking through projection matrices, knowing the 3D points of the reference images. This is based on the work of Hartley, R. and Zisserman, A. [Har00] about factorization process of the fundamental matrix, which encapsulates the epipolar geometry and the essential matrix ($E = R \times T$), in this case, by decomposition, is projection or transformation matrices that relates the images and the coordinate system of the world always relative to the right image or second image. A more practical work is presented in Ma, Y. et al. [Ma06], which works directly with calibrated points and achieve compliance by the homographies and projection matrices that has reasonable mistakes, which are optimized for use later.

The purposes of this work focus on developing a methodology and propose an automatic architecture allowing insert annotations and virtual objects, without fiducial marker or intervention of the environment, near to the object in the showcase exhibited in the Archaeological Museum San Miguel de Azapa.

2. SYSTEM ARCHITECTURE

The system input consists of a sequence of images from the camera mounted on the video-based HMD or mobile device (PDA or Cellular). The system is made up of four main modules, where the former is responsible for selecting the object and its recognition by the method reported in [Her06]. If the object exit at the database the process continues with the initialization of the tracking. The next image of the sequence is considered second in order to

initialize the tracking cycle, determining the correspondence matching using Scale Invariant Feature Transforms (SIFT) [Low04], where the points of interest of the recognized object is fit to the points identified in the second image, the third module calculates the homography with the corresponding points resulting from the matching and is optimized by robust calculation; and then selected points of entry are projected. Tracking to the next image consists of detecting image points in next image, then performing the matching with the points of the previous image. If the resulting number of matching points is greater than four points, then tracking process continues. The fourth module is responsible for positioning and inserting the virtual object so that it does not obstruct the object of interest (see Figure 1).

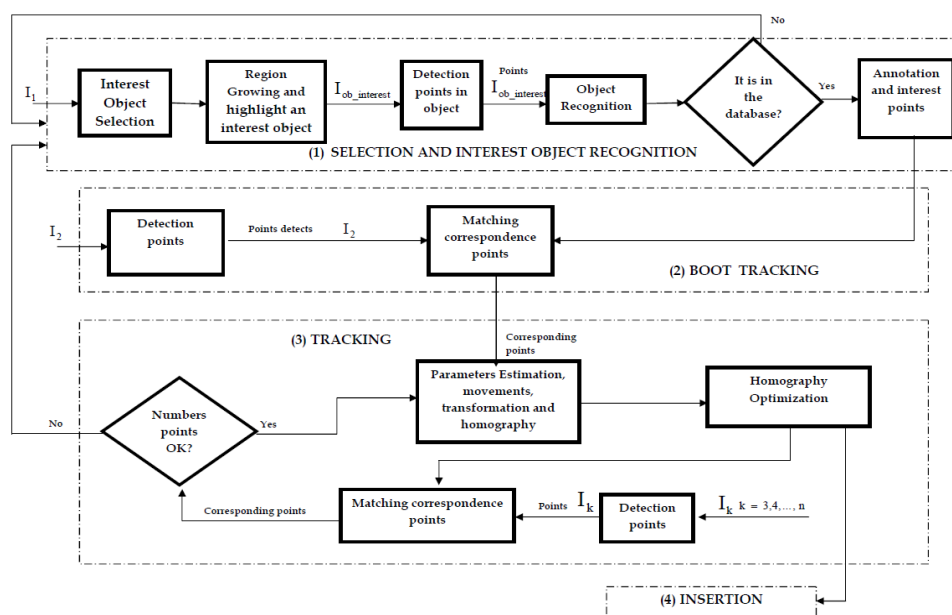


Figure 1. Proposed Architecture of the Augmented Reality System.

3. RESULTS

The test platform used to carry out the proposed method was a laptop with the following features: Intel Centrino Core 2 Duo T5550 processor, 1.8 GHz (2 MB cache, 3 GB of RAM DDR-2) with Windows XP operating system. The algorithm implementation was made using Visual Studio 2005 y Matlab Compiler, due to its ease of use and data management.

Following, the results step by step obtained during the development of the application is presented.

Selection and identification of the object of interest

In Figure 2a, is shown the first image, in which you select the object of interest (Textile loom) and is enhanced using the region growing algorithm (see

Figure 2.b). From this new image (Figure 2.b), proceeds the detection of characteristic points to recognize the object of interest and thereby seek the associated annotation on the database, then proceed to add the annotation (see figure 2.c.).

Initialization of Tracking

Once recognized the object of interest, initialization of tracking is performed, in order to detect characteristic points of the second image and perform correspondence between the selected points of interest from the object and those detected on the second image.

Tracking

After the correspondence of points, we can see there are points that are not relevant to the object of

interest, for this reason we proceeded to remove these



Figure 2a. Image where the object of interest is selected.



Figure 2b. Object of interest.



Figure 2c. Annotation associated to the object of interest.

From these points we have the conditions to insert the annotation associated to the object of interest, using the process described in the system's architecture.

To verify the proposed method of tracking one object of interest is representing in showcase 6 of Archaeological Museum San Miguel de Azapa, which in this case is "Ceremony Helmet". The result of these tracking is found in Figure 3.

Effectiveness of the System

To analyze the results, we will consider only a showcase of the Archaeological Museum San Miguel de Azapa, in this case, we use the showcase 6 (see Table 1). This case includes the following items.

To perform the tests objects that are more representative here used. These objects are 1, 2, 14 and 16.



Figure 3. Test: tracking objects "Ceremony Helmet", courtesy of Archaeological Museum San Miguel de Azapa, Arica Chile.

N°	Objects	N°	Objects	N°	Objects	N°	Objects
1	Ceremony Helmet	5	Tray of Cociba	9	Points of Harpoons and fishing tools	13	Ceramic Pitcher
2	Textile Loom	6	Ceremony Plate	10	Cave painting	14	Ceramic Vase
3	Fishermen Pocillo	7	Kitchen Tools	11	Mummified skull	15	Mini pitcher
4	Fishing string	8	Fishing Harpoons	12	Rudimentary fishing harpoon	16	Typical Dress

Table 1. Objects in showcase 6 of the Archaeological Museum San Miguel de Azapa.

An important aspect to consider is the time it takes to make the insertion process of the annotation associated with the object of interest, as this process should be performed in real time.

In table 2 then assesses the time (in milliseconds) it takes to make the insertion process. This will take 5 iterations of the process.

As can be seen in Table 2, the average time it takes the system to perform the insertion of entries is 27 milliseconds, which is below the time required to make it in real time.

In the test, the results are valid for real-time or appropriate time en mobiles applications [Wag05], according to the time it takes to make integration meet the requirements.

		Time (milliseconds)					
	Object Showcase 6	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5	Average
1	Ceremony Helmet	28,74	31,63	22,94	24,10	28,32	27,14
2	Textile Loom	19,64	19,93	27,38	20,05	24,61	22,32
14	Ceramic Vase	10,18	6,51	9,48	6,45	6,11	7,75
16	Typical Dress	20,89	26,51	27,70	16,25	28,36	23,94

Table 2. Time (in milliseconds) to delay the insertion of annotations for objects 1, 2, 14 and 16.

4. CONCLUSIONS

The exploration of this area of image recognition and application of augmented reality in a space where the viewer is limited in connection with the object appears as a viable alternative and a contribution to multiple works on the use of PCV in computer vision.

According to the results, we conclude that the objects in the showcase number six are recognized without any problem, regardless of the size the region understands; this is due to the SIFT point detection and its use in establishing the automatic therefore it was used in earlier work of Harris and matching detection Pilu, see [Her06].

From the standpoint of characteristics points detector, it can be concluded that it must not only recognize a lot of points, but also that these are the most representative objects of interest, so that when carrying out the correspondence between images the object of interest and the second image successively obtained their positions. It is therefore very essential to optimize the points corresponding to eliminate those that do not belong to the object in question, and so, determine a best homography. The problem that arises is that for small objects, making this optimization eliminates points, which in some cases, prevents the determination of homography.

From the viewpoint of the insertion of objects, their accuracy depends heavily on the preliminary processes and taking advantage of existing natural brands in the showcase for the determination of the position. From the results obtained, we can conclude that for objects of considerable size, the insertion of annotations with a proper position perform a hundred percent, this was not so for small objects because of the number of points detected.

Regarding the time that it takes to perform the insertion of the annotation, we can conclude that the times obtained in the tests, are below those required to be used in real time, so the proposed method is valid to be used in real-time or appropriate time in mobiles applications.

5. FUTURE WORK

Performing the evaluation from the point of view of usability of the final system integration and insertion of alluding audio to the object on display will be considered. From the standpoint of computer vision operational comparisons with predictive and adaptive filters will be considered.

6. ACKNOWLEDGMENTS

This research has been partially financed by Grant 8763-by University of Tarapacá, Arica, Chile.

Special thanks to Dr. Julia Cordova Gonzalez from MASMA of UTA, for her support and archaeological knowledge of Andean history.

7. REFERENCES

- [Bil06] Billinghurst, J. (2006), "Augmented Reality Applications". [On-Line]. Dirección URL: <http://www.newhorizons.org/strategies/technology/billinghurst.htm>.
- [Bro96] Brooks, F. The Computer Scientist Toolsmith II. CACM 39, 3 p.p. 61-68, 1996.
- [Har00] Hartley, R. and Zisserman, A. Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge, 2000.
- [Her06] Herrera-Acuña, R. and Aracena-Pizarro, D. Image recognition and retrieval using an invariant schema based on principal component vector analysis, (IeCCS2006) E-Conference for Computer Science, for publishing in Lecture Serie on Computer and Computational Sciences ISSN 1573-4196, Grecia, 2006.
- [Low04] Lowe, D. Distintive Image Features From Scale Invariant Keypoints. University Of British Columbia, Canada, 2004.
- [Ma06] Ma, Y., Soatto, S., Kosecka, J. and Sastry, S. An Invitation to 3-D Vision: From Images to Geometric Models. Springer, 2006.
- [Mal02] Malik, S., Roth, G., McDonald, C. Robust 2D Tracking for Real-time Augmented Reality. In Proceedings of Vision Interface (VI), Calgary, Alberta, Canada, p.p. 399-406, 2002.
- [Sim00] Simon, G., Fitzgibbon, A. and Zisserman, A. Markerless Tracking using Planar Structures in the Scene. Proc. Int'l Symp. Augmented Reality (ISAR 2000), IEEE Computer Soc. Press, Los Alamitos, Calif., p.p. 120-128, 2000.
- [Vil06] Vilars. Sistema de Realidad Aumentada Vilars. 2006, [On-Line] Dirección URL: <http://griho.udl.es/i2004/vilars.htm>.
- [Wag05] Wagner, D., Pintaric, T., Ledermann, F. and Schmalstieg, D. Towards Massively Multi-User Augmented Reality on Handheld Devices. Third International Conference on Pervasive Computing (Pervasive 2005), May 9-10, Munich, Germany, 2005
- [Yua06] Yuan, M., Ong, S. and Nee, A., Registration Using Natural Features for Augmented Reality Systems. IEEE Transactions on Visualization and Computer Graphics, Volume 12, 4, p.p. 569 - 580, 2006.

Implementing the Local Binary Patterns with SIMD Instructions of CPU

Roman Juránek, Pavel Zemčík, Adam Herout

{ijuranek,zemcik,herout}@fit.vutbr.cz

Graph@FIT

Department of Computer Graphics and Multimedia

Faculty of Information Technology

Brno University of Technology, Brno, Czech Republic

ABSTRACT

Usage of statistical classifiers, namely AdaBoost and its modifications, is very common in object detection and pattern recognition. Performance of such classifiers strongly depends on low level features they use. This paper presents an experimental implementation of the Local Binary Patterns (LBP) that uses SIMD instructions for acceleration. The experiments shows that the proposed implementation is about six times faster than the plain C implementation (i.e. with no special optimizations) and superior to optimized implementations of features with similar descriptive power.

Keywords: LBP, AdaBoost, Object Detection, Feature Extraction, SIMD, SSE, CUDA

1 INTRODUCTION

Object detection in images and video sequences has wide range of applications. Several object detection methods exist; however, one of the best available methods today is exploitation of statistical classifiers. The statistical classifiers are able to distinguish an object from non-object in a small window. To detect the object in image data or video, it is necessary to scan the image or video frame and apply the classifier to each possible window location within the scanned image. The classifier works with low-level features extracted from the classified image window. The features are usually simple functions of selected pixels from the window. The design of the features significantly affects the performance of the classifier and its speed and thus the speed of detection.

The Local Binary Patterns (LBP) [7] discussed in this paper are widely known to be good features to describe local areas. They are frequently used in texture analysis and segmentation. The recent studies show that they are usable as features for classification as well.

In this paper, the high performance implementation of LBP feature extraction is introduced. The implementation exploits the SIMD instructions (namely SSE) available on contemporary CPUs. The implementation is comparable with similar implementations of other feature types – Local Rank Differences (LRD) [13] and Local Rank Patterns (LRP) [6]. For the experiments

WaldBoost (a modification of AdaBoost [11]) training algorithm [9] has been used.

2 RELATED WORK

The interest in fast feature extraction has been earlier pursued by other researchers because the extraction speed influences the overall performance of the object detection. Efficient implementation is thus necessary for practical applications of object detection.

Innovative approach that constituted a breakthrough in real-time object detection was taken by Viola and Jones [11]. They converted the input image into an integral representation which allows for calculation of Haar features in constant time.

There has also been much effort to implement the AdaBoost based object detection on hardware platforms like FPGA or ASIC chips [12, 10, 5]. Typically they use traditional Haar features. Recently there were proposed implementations of object detection with Local Rank Differences image features (LRD) on GPU [8] and CUDA [2] which employs resources of modern graphics cards to accelerate feature extraction. The disadvantage of this approaches is that they still need special hardware.

In our previous work, we introduced high performance implementation of the Local Rank Differences [1] and Local Rank Patterns [4] image features. These implementations exploited the SIMD instructions of Intel CPU. This work shares same framework with the mentioned implementations of LRD and LRP.

3 LOCAL BINARY PATTERNS

Local Binary Patterns [7, 14], in their basic form, capture information about local textural structures through thresholding samples from local neighborhood by its central value. From the thresholded values a pattern

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

code is formed such that each sample is represented by a single bit (Figure 1). The image pixels or response of some sampling function can be used as the samples. In many applications, circular neighborhood with 8 samples is used (8 bit pattern).

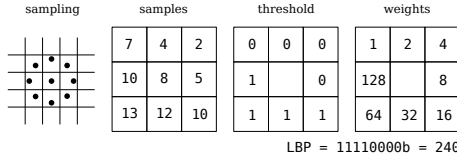


Figure 1: Evaluation of the LBP feature. First the samples are taken from the image. The values are then thresholded by the central value.

In this form, the feature response is dependent on the feature orientation. In some applications, such as texture classification or image segmentation, the rotational invariance is needed. In this case, the feature response is normalized (e.g. by bit shifting).

In our approach, the feature consists from 3×3 regularly spaced rectangular samples. The classifier can hold feature instances of different sizes. However, for practical reasons the size of a sample is constrained to maximum of 2×2 pixels. And therefore four possible samplings exist as shown in the Figure 2. This constraint, as we show in [4], does not introduce any reduction of classifier precision.

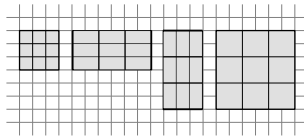


Figure 2: Possible configurations of LBP features.

4 LBP EVALUATION

The traditional way of the feature evaluation is to gather samples from the input image and sequentially construct the LBP code bit by bit. The evaluation in the proposed implementation is optimal in two ways. First, it minimizes number of memory accesses by using pre-calculated image representation – interleaved convolution image [1, 4], which allows for fast retrieval of necessary data. Second, the evaluation takes advantage of the SIMD instructions by processing all values of the feature in parallel manner.

4.1 The SSE Instruction Set

Before the actual implementation of the LBP feature evaluation is described, let us briefly characterize the SSE instruction set on Intel CPUs. Unlike classic x86 instructions where an operation is executed on one piece of data at the time, the SSE set provides means to execute one operation over multiple data (as shown in the Figure 3). Therefore the main attribute of the SSE instruction set is parallel processing of data.

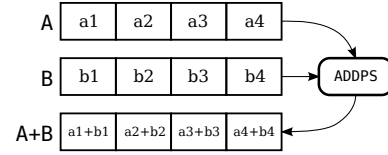


Figure 3: Operation of SSE instruction ADDPS. The A and B are considered to be vectors of four float values.

The SSE set was introduced in 1999 as an extension to x86 and MMX instruction set. The instructions works with 128 bit wide registers which can hold a vector of values (e.g. 4×32 bit float, 16×8 bit integer, etc.).

4.2 Image Preprocessing

As was previously stated, in our approach we use rectangular samples as an input for feature evaluation. In the preprocessing stage, images convolved with all possible shapes of samples are created. This images are later used as source of samples. Four images are created in our case as the size of the samples is restricted to maximum of 2×2 pixels.

Every convolved image is arranged in a manner that four consequent pixels in the memory (i.e. 32 bit word as we use 8 bit images) correspond to 2×2 adjacent convolution responses. This memory layout requires that each convolved image is logically divided into blocks where each block contains sub-sampled image convolved with same modulo position of convolution kernel. Number of blocks is determined by $w \times h$ where w and h is width and height of the convolution kernel. This is schematically shown in Figure 4 where preprocessing of image with 2×1 pixel kernel is displayed and two memory blocks are formed. Images preprocessed with other kernels are created analogically. This layout ensures that data for each feature are placed in same block and can be obtained by two memory accesses.

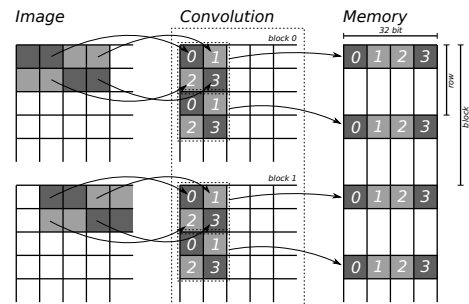


Figure 4: Preprocessing of image with 2×1 pixel kernel.

Note that the feature evaluation itself is independent on the choice of the convolution kernels. The rectangular shape was selected because the implementation

of convolution and memory rearrangement can be very highly optimized.

4.3 The Evaluation

The framework of evaluation of the features is displayed in the Figure 5. The input image is first pre-processed – the convolution images are created. Each feature is parametrized by a position in the image and size of samples. The convolution image is selected according to the size of samples and the block and address within the block is determined from the feature position.

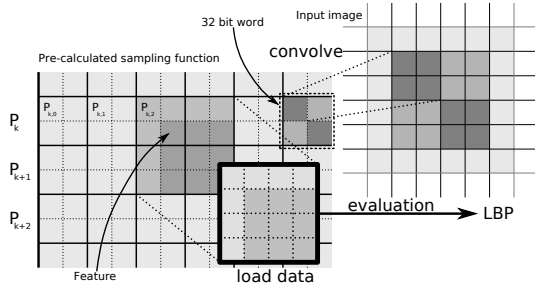


Figure 5: Schematic view of the image preprocessing and LBP evaluation. In this case a feature with 2×2 pixel samples is used.

By loading eight 32 bit aligned pixels from two subsequent rows of a convolution image (i.e. two 64 bit reads), the 4×4 responses of the sampling function are effectively loaded. The feature data is then located in a 3×3 sub-window of this data (feature shift). Note that the feature evaluation code is *independent* on the size of the feature as the data are loaded from precalculated representations.

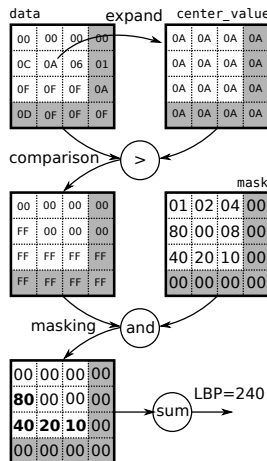


Figure 6: Block diagram of LBP evaluation.

Figure 6 shows the feature evaluation step by step. The corresponding code is displayed in the Figure 7. Every grid represents a 128 bit SIMD register holding sixteen 8 bit values loaded from the pre-calculated sampling function. First, the central value (defined by feature shift) is expanded to the full register width and

compared with all other values. The comparison result then serves as a mask for vector of weights (again selected according to the feature shift). The masked weights (i.e. weights for which the comparison resulted in *true*) are then summed up producing the LBP value. Note that, although the comparison can result in *true* on positions that do not belong to the feature, this will not influence the result as the weights are always *zero* for them.

```
union {
    __m128i q;
    signed short ss[8];
} result = {
    _mm_sad_epu8(
        _mm_and_si128(
            lbpWeights[shift].q,
            _mm_cmpgt_epi8(
                data,
                _mm_set1_epi8(*center))),
        zero)
};
int lbp = result.ss[4] + result.ss[0];
```

Figure 7: The actual evaluation code (with Intel intrinsic functions). The `lbpWeight[shift]` selects one of four possible weight vectors, `data` is vector of the data and `center` is the pointer to feature central value.

5 RESULTS

The experiments were conducted on a PC with Intel Core i7 CPU (eight core), 4 GB DDR3 and CUDA capable ASUS NVidia ENGTX 280 graphics card.

The proposed implementation was compared to experimental implementation of LBP on CUDA architecture [3] and to the implementation with no special optimizations (which we refer to as *Plain C*). And also compared to implementations of the LRD and LRP extractors which shares same evaluation framework.

The implementations were tested on the task of multiscale face detection. In the first test, we compare the real detection performance in terms of processed frames per second on a long video (taken from public TV broadcasting).

$\alpha = 0.1$	$560 \times 240px$	$720 \times 576px$	$1280 \times 720px$
SIMD	61/58/57	22/20/20	10/10/10
CUDA	88/73/69	68/56/54	27/24/23
Plain C	8/6.4/5.6	3.4/3.5/2.8	1.4/1.3/1.1
$\alpha = 0.2$	$560 \times 240px$	$720 \times 576px$	$1280 \times 720px$
SIMD	87/82/81	28/24/24	13/11/11
CUDA	115/91/89	82/63/61	31/27/26
Plain C	12/10.4/9.6	4.5/4/3.7	1.9/1.6/1.5

Table 1: Object detection performance in *frames per second* on different architectures with usage of different feature extractors. The values in the table are ordered in following way: LBP/LRP/LRD.

Results in Table 1 display performance of processing of three videos with different resolutions ranging

from low resolution to 720p HD video. Two classifiers with different target error rate (α) were used. The experiment shown that the SIMD implementation outperforms the Plain C. On the low resolutions the SIMD almost reaches the performance of the multiprocessor CUDA implementation although on the high resolutions the CUDA is almost three times faster.

	560 × 240px	720 × 576px	1280 × 720px
SIMD	0.32	0.98	2.1
CUDA	0.29	0.43	0.82
Plain C	2.5	7.8	17.5

Table 2: Preprocessing performance [ms/frame]

The Table 2 shows comparison of preprocessing on the three architectures. The preprocessing includes image scaling to build image pyramid. Additionally, in the SIMD implementation, each pyramid level is convolved according to description in the Section 4.

6 CONCLUSION AND FUTURE WORK

The goal of the work presented in this paper was to efficiently implement the Local Binary Patterns feature extractor in the contemporary CPUs using the advanced SIMD instructions. The goal was fulfilled and high performance feature extractor was implemented and experimentally evaluated.

The extractor needs preprocessing stage which prepares convolution images with precalculated sampling functions. In the evaluation stage, each feature needs only two memory accesses to load necessary data and the data are then processed by SIMD instructions to calculate the LBP feature response.

The results show that the SIMD extractor is faster than the Plain C extractor by factor of 6.5 and almost reaches to performance of implementation on CUDA architecture. And the speed of LBP evaluation is comparable or better than evaluation of feature types as LRD and LRP. Although the CUDA implementation is faster, it needs special hardware which supports the CUDA architecture. On the other hand, the SIMD instructions are common in all contemporary CPUs which makes the SIMD implementation suitable for larger range of applications. Especially for embedded systems.

The future work includes optimization of feature evaluation on bulk data, for example evaluation of several independent features in parallel or simultaneous evaluation of consequent features. The future work also includes further dataflow optimizations.

ACKNOWLEDGEMENTS

This work was supported by Czech Ministry of Education, Youth and Sports research project Center of Computer Graphics (LC06008).

REFERENCES

- [1] Adam Herout, Michal Hradiš, Roman Juránek, and Pavel Zemčík. Implementation of the "local rank differences" image feature using simd instructions of cpu. In *Proceedings of Sixth Indian Conference on Computer Vision, Graphics and Image Processing*, page 9, 2008.
- [2] Adam Herout, Radovan Jošth, Pavel Zemčík, and Michal Hradiš. Gp-gpu implementation of the "local rank differences" image feature. In *Proceedings of International Conference on Computer Vision and Graphics 2008*, Lecture Notes in Computer Science, pages 1–11. Springer Verlag, 2008.
- [3] Adam Herout, Radovan Jošth, Pavel Zemčík, and Michal Hradiš. Gp-gpu implementation of the "local rank differences" image feature. In *Proceedings of International Conference on Computer Vision and Graphics 2008*, Lecture Notes in Computer Science, pages 380–390. Springer Verlag, 2008.
- [4] Adam Herout, Pavel Zemčík, Michal Hradiš, Roman Juránek, Jiří Havel, Radovan Jošth, and Martin Žádník. *Low-Level Image Features for Real-Time Object Detection*, page 25. IN-TECH Education and Publishing, 2009.
- [5] M. Hiromoto, K. Nakahara, H. Sugano, Y. Nakamura, and R. Miyamoto. A specialized processor suitable for adaboost-based detection with haar-like features. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, June 2007.
- [6] Michal Hradiš, Adam Herout, and Pavel Zemčík. Local rank patterns - novel features for rapid object detection. In *Proceedings of International Conference on Computer Vision and Graphics 2008*, Lecture Notes in Computer Science, pages 1–2, 2008.
- [7] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(7):971–987, 2002.
- [8] Lukáš Polok, Adam Herout, Pavel Zemčík, Michal Hradiš, Roman Juránek, and Radovan Jošth. "local rank differences" image feature implemented on gpu. In *Proceedings of the 10th International Conference on Advanced Concepts for Intelligent Vision Systems*, Lecture Notes In Computer Science; Vol. 5259, pages 170–181. Springer Verlag, 2008.
- [9] Jan Sochman and Jiri Matas. Waldbost – learning for time constrained sequential detection. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, pages 150–156, Washington, DC, USA, 2005. IEEE Computer Society.
- [10] T. Theodorides, N. Vijaykrishnan, and M.J. Irwin. A parallel architecture for hardware face detection. In *Emerging VLSI Technologies and Architectures, 2006. IEEE Computer Society Annual Symposium on*, volume 00, pages 2 pp.–, March 2006.
- [11] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1:511–518 vol.1, 2001.
- [12] Yu Wei, Xiong Bing, and C. Chareonsak. Fpga implementation of adaboost algorithm for detection of face biometrics. In *Biomedical Circuits and Systems, 2004 IEEE International Workshop on*, pages S1/6–17–20, Dec. 2004.
- [13] Pavel Zemčík, Michal Hradiš, and Adam Herout. Local rank differences - novel features for image. In *Proceedings of SCCG 2007*, pages 1–12, 2007.
- [14] Lun Zhang, Rufeng Chu, Shiming Xiang, ShengCai Liao, and Stan Z. Li. Face detection based on multi-block lbp representation. In *ICB*, pages 11–18, 2007.

Description of image content by means of graph grammars

Jiří Zuzaňák, Aleš Láník, Pavel Zemčík
Graph@FIT

Department of Computer Graphics and Multimedia
Faculty of Information Technology
Brno University of Technology, Brno 612 66, Czech Republic
{izuzanak,ilanik,zemcik}@fit.vutbr.cz

ABSTRACT

This paper presents an idea for partial bottom-up parse of image content by use of an attributed graph grammar, in order to achieve effective high-level representation of knowledge contained in image. Terminal nodes of the proposed grammar are formed by image objects (points, lines, and objects detected by classifiers) and areas detected in image by various image processing and segmentation methods. Based on attributes of terminal nodes, each production rule creates derived attributes for high-level representation of lower-level knowledge. Graph that is parsed by graph grammar is constructed in process of knowledge extraction by application of segmentation and image processing algorithms. Created graph is then processed by sequential application of graph grammar rules. Left side of rules is detected by isomorphism detector, and consequent rewrite is performed by rule with highest priority. A part of rewrite process is represented by processing of evaluations of vertices and edges, that describe various properties of objects and their relationships. Further in the paper we present example of attributed graph grammar application in order to describe image content.

Keywords: Graph grammar, Rewriting system, Bottom-up graph analysis, Knowledge representation

1 INTRODUCTION

Methods of region, object and edge detection forms basic approaches exploitable in process of image analysis. Mentioned methods produces set of objects and regions, with describable relations between them. For example, we can say that the detected edge a is parallel with edge b and has common end point with edges c and d .

Such information can be easily represented by structured data. In fact, in image analysis and pattern recognition, data structures are used for representation of objects topology and relations between these objects, which are not in basic properties different from mathematic graphs. Based on this fact it can be said that attributed graphs with unrestricted possibilities of vertex and edge evaluations can describe any of these structures.

Graphs created by the outlined approach are complicated structures, with dense set of vertices and edges, which cannot be easily interpreted without further modification. Principle of graph grammars based on graph rewriting systems (which will be further introduced)

can be successfully applied in sense of reduction complexity of structured data represented by graph. Graph grammars thus enable creation of effective representation of knowledge contained in processed image. This paper targets extraction of knowledge contained in image (represented by graph of objects from which is image composed) by application of graph grammar rewriting system.

The paper is organized as follows. In section 2. papers and publications dealing with similar problem are mentioned. In section 3. the proposed approach to description of knowledge represented by image is introduced. Further in this section, an example of graph grammar and its application to image description is presented. Section 4. describes actual state of work on graph grammar rewriting system implementation and image processing tools. Finally section 5. concluding paper, contains the results achieved so far and proposed ideas for consequent work.

2 RELATED WORK

The idea of using graph grammars or some other formalism for recognition of image content described by structured data is not revolutionary. Many efforts based on use of attributed grammars, rule-based analyzers, and rewriting systems for image description have been made.

In [11], K. C. You and King-Sun Fu, describe object shape by attributed grammar, where its terminal symbols denote open curve segments and angle between two adjacent segments. Feng Han and Song-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Chun Zhu propose in [4] grammar composed from six rules, which enable (based on edges detected in image) description of rectangular objects and arrays of rectangular objects in image by use of top-down/bottom-up inference.

Strong image recognition and description system was introduced in [8] by Yuichi Ohta. The system is based on segmentation of input image by various methods, and sequential combined top-down and bottom-up analysis of retrieved information against model data. The introduced approach is not purely graph grammar rewriting system but more likely (as authors say) rule-based region analyser. Similar approach is proposed in [7] by Yuichi Ohta, Takeo Kanade, and Toshiyuki Sakai. Input image is segmented to basic areas by testing of intensity data, which are described by structured symbolic data. The knowledge represents set of rules by semantic nets.

A model called spatial random tree grammars is introduced in [10] by J. M. Siskind, Jr J. Sherman, I. Pollak, M. P. Harper, and C. A. Bouman. The approach is based on Bayesian methods. The authors developed algorithm for exact computation of likelihoods and maximum a posterior probabilities, and EM updates for model parameters estimation. The method is applied to the task of classifying of natural images and it is shown that use of described hierarchical structure significantly improves classifier effectivity.

In [3], an approach for image content description is proposed by T.-J. Fan, G. Medioni, and R. Nevatia, based on detection of image objects and finding relations defined between them. Introduced examples are based on description of 3D objects by their surfaces detected in input image by segmentation method based on curvature properties. Detected objects and their relations are in result represented by graph, whose vertices represent patches and edges represent geometric relations between them. Inference of objects in these graphs is then performed by reasoning of the type of connections between adjacent patches. The used graph processing is not based on graph grammars or some formal graph rewrite system.

The authors, Chungan Lin and Ramakant Nevatia, propose in [5] method for detecting of buildings and description of their 3D shape by use of geometric and projective constraints. These constraints are used to generate hypotheses for the presence of building roofs from low-level linear features, and their parallelograms. Generated hypothesis are then verified against created models.

3 PROPOSED APPROACH

In this section is on practical example demonstrated application of graph grammar rewriting system for detection of crosswalks in image. The proposed approach is based on detection of edges and, consequent creation of

simple graph describing detected edges and their relations.

3.1 Image processing

Detection process is composed from the following steps: In source image, whose example is displayed in Figure 1, edges are detected by standard algorithms (Sobel, Laplace, Laplacian of Gaussians), displayed in Figure 2. Detected edges are then interpolated by lines described by geometric parameters (Hough transform). The result is displayed in Figure 3.



Figure 1: Crosswalk original



Figure 2: Crosswalk edges

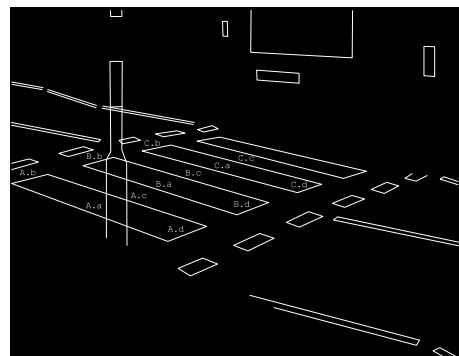


Figure 3: Crosswalk interpolated strong edges

The created set of edges is then preprocessed in order to retrieve their relations, such as common points, intersection points, orthogonality, shortest and longest distance, etc., depending on needs of defined graph gram-

mar and its rules. Based on retrieved information is then created graph, whose vertices describe image edges, and edges of the graph describe relations between these vertices.

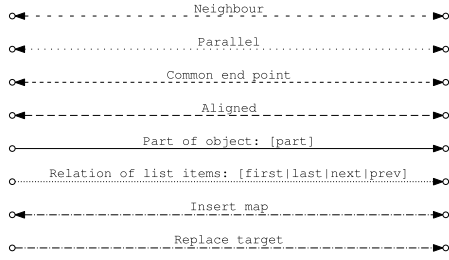


Figure 4: Edge colors legend

In image graph, are by edges expressed the following relations: Neighbour - connected objects are close each to other and there are no other objects between them. Parallel - connected edges are parallel. Common end point - connected edges have at least one common end point. Aligned - connected objects are aligned. Part - target object is part of source object. Tile list relation - description of relations between items in list object.

Graph created from input image (describing only labeled edges from Figure 3.) is displayed in Figure 5. Legend denoting color and style of edges used in example graphs is displayed in Figure 4.

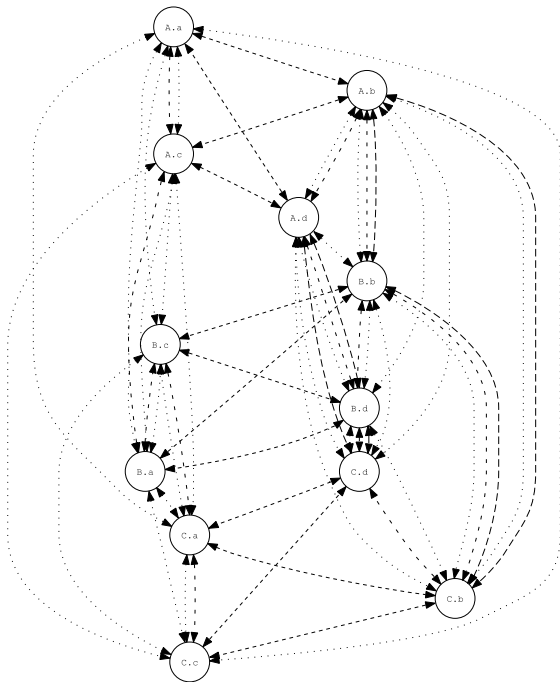


Figure 5: Graph of edges from image in Figure 3.

3.2 Example of graph grammar

Simple graph grammar designed for description of image content is composed from seven rules. Basic principle of the presented grammar is detection of tiles,

and recursive merge of detected tiles to linear structure describing crosswalk. Short explanations of each grammar rule and their representation by graph follows. Number in brackets denote rule priority (rule with lower priority precede before rule with higher priority).

First rule (Figure 6.) serve for detection of tiles, based on description of tile by four edges connected in end points, where opposite edges of tile must be parallel. Second rule (Figure 7.) adds neighbour property to tiles determined by neighbour properties of edges from which are these tiles composed. Third rule (Figure 8.) mark tiles as aligned by detection of their mutual aligned edges. Rule determines alignment of tiles only when these tiles are already in neighbour relation. Fourth rule (Figure 11.) describes initialization of tile list. Tile list serves as linear list of aligned tiles where each two adjacent tiles are neighbour in processed image. Described structure serves as basic tool for description of crosswalk structure. Fifth and six rules (Figures 10. and 11.) describe insertion of new neighbour tiles to begin or end of already existing list of tiles. Last rule displayed in Figure 12. describes simple rewrite from list of tiles vertex to crosswalk vertex, which determine final detection of crosswalk. Recognize of incomplete tile lists is disabled by setup of rule priorities.

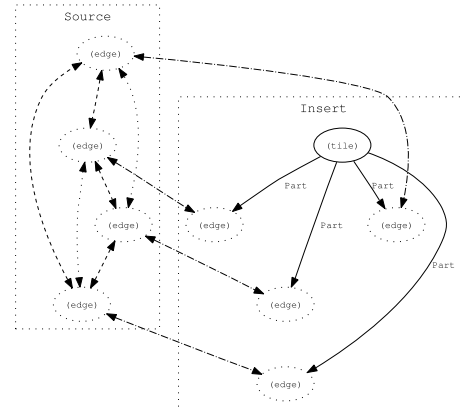


Figure 6: Rule 1: Detection of Tile (0)

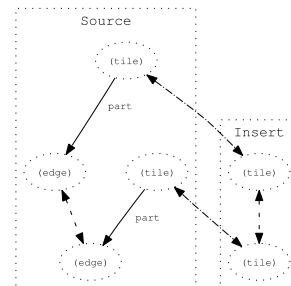


Figure 7: Rule 2: Neighbour tiles (1)

Important property of mentioned rules is their priority. Priority of rule determines order, in which the rules are applied when possibility exists to rewrite by more

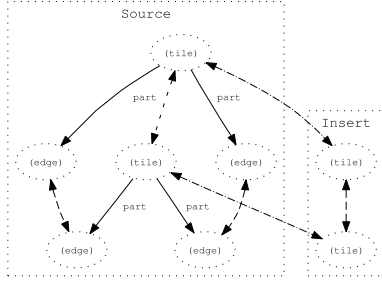


Figure 8: Rule 3: Aligned tiles (2)

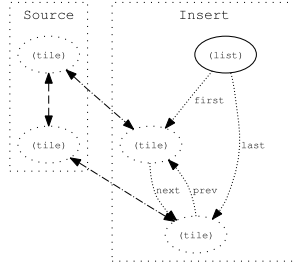


Figure 9: Rule 4: Initialization of tile list (4)

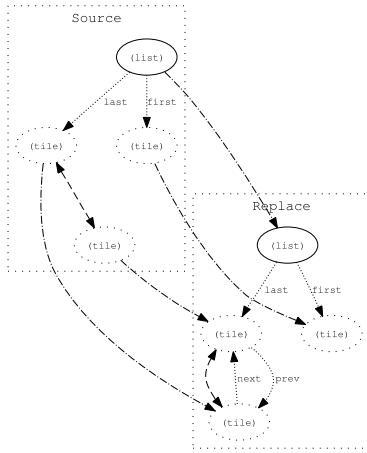


Figure 10: Rule 5: Insertion to tile list end (3)

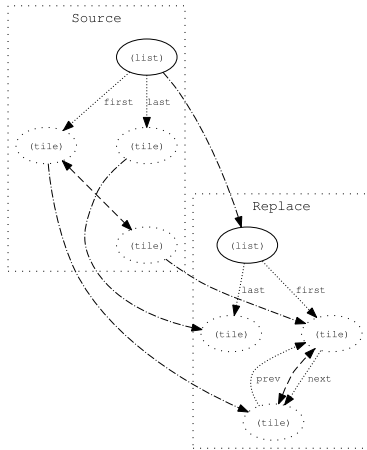


Figure 11: Rule 6: Insertion to tile list begin (3)

than one rule. Good example of rule priority importance are rules 6 and 5 vs. rule 7, ensuring that if there are some crosswalk tiles that can be inserted to tile list then rewrite by rule 7 is disabled.

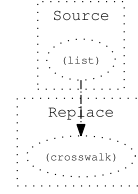


Figure 12: Rule 7: Detection of crosswalk (5)

Proposed graph grammar detects standard crosswalks composed from orthogonal tiles i.e. continental type of crosswalk displayed in Figure 13. By creation of special grammar for each type of crosswalk, we are able to generalize (by proper rules) different crosswalks to one general class.

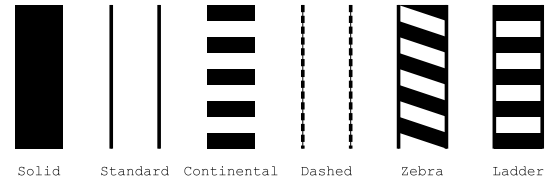


Figure 13: Different types of crosswalks

4 ACTUAL STATE OF WORK

Graph grammar processing tool is in state of development. This tool consist from several components, which are: graph representation, graph isomorphism detector, graph rewriting system, and graph grammar representation.

Component for representation and manipulation with graphs is complete. This tool enables loading and saving of graphs with arbitrary structure and standard vertex and edge evaluations (standard data types, strings, arbitrary data types controlled through dynamic libraries).

Abstract data type set is implemented as red-black trees, enabling fastest possible searching for processed vertices and edges. Graph representation enables execution of various graph algorithms as are: detection of spanning tree, search for shortest path, detection of isomorphism between two graphs, decomposition to graph components, and more graph algorithms.

Second component of graph grammar processor is represented by graph isomorphism detector. Created detector of graph isomorphisms is specially designed for application in graph grammar processing system. Detector is based on automatized creation of so called graph parser (graph automata) which is intended for searching of set of subgraphs isomorphisms in one host graph. Isomorphism detector use described representation of graphs, which enables dynamic modification of graphs and thus is proper for graph rewriting.

Graph rewriting system which forms third part of graph grammar processing tool is in phase of design and partially in phase of implementation. The rewriting system so far enables loading of graph grammars

(set of rules with defined priorities, where each rule is composed from two graphs, and transformation relations between these two graphs), creation of graph parser described previously (based on left side of grammar rules), detection of left rules and basic rewrite, so far without more complicated processing of vertex and edge evaluations.

Last part of graph grammar processor is formed by graph grammar representation, and it is in phase of design and implementation similarly as previous component. This component is closely bind to graph rewriting system.

In current state, graph grammar processing tool is capable of generating graphs based on set of given graph grammar rules, and parsing of these generated graphs by grammar reverse to generating grammar. Experimental grammars for processing graphs of program flow has been created and tested. From the results it can be implied that designed graph rewriting tool based on defined graph grammar is capable to significantly optimize graph of program flow.

5 CONCLUSION

In this paper was proposed an idea for description of image content by graph grammar rewriting system, along with example of graph grammar of such system applied for description of simple image content.

Content of an image is in the first steps of process transformed to graph, which describes objects of image by graph vertices, and relations between these objects by graph edges. Further information describing image content is represented by evaluation of graph vertices and edges. The created graph is then interpreted by graph grammar processor, where individual rewriting steps determine structure of data represented by graph. Applied graph grammar processor is based on rewrite system which is in phase of design. Rewrite system is based on graph isomorphism detector mentioned in Section 4.

Graph grammar processing tools has been so far tested on grammars generating graphs of program flow and reverse grammars parsing generated graphs. From the achieved results it can be seen that graph rewriting system is capable of described properties (parsing and interpreting complicated graph structures).

Further work will concentrate on finishing design and implementation of grammar representation and graph rewriting system. Next steps in design of graph isomorphism detector and consequently graph rewriting tool will be:

- Complex evaluation of vertices and edges, and mainly their processing and configurable modification by process of graph rewrite

- Allow use of stochastic graph grammar rules (designed if possible without need of change of graph isomorphism detector)
- Integration of graph rewriting system to graph grammar description, and completion of system allowing complex graph analyses based on given graph grammar
- Testing of designed system and evaluation on data extracted from real world images

Essential step of whole process will be testing of designed system on real world data, retrieved from videos and images provided by European project WeKnowIt, of which is this work part.

ACKNOWLEDGEMENTS

This work is supported by the European Commission under contract FP7-215453 - WeKnowIt.

This work was (also) supported by the project "Security Oriented Research in Information Technology" by Ministry of Education, Youth and Sports of Czech Republic no. MSM0021630528.

REFERENCES

- [1] L. P. Cordella, P. Foggia, C. Sansone, and M. Vento. An improved algorithm for matching large graphs. In *In: 3rd IAPR-TC15 Workshop on Graph-based Representations in Pattern Recognition*, Cuen, pages 149–159, 2001.
- [2] D. G. Corneil and C. C. Gotlieb. An efficient algorithm for graph isomorphism. *J. ACM*, 17(1):51–64, 1970.
- [3] T.-J. Fan, G. Medioni, and R. Nevatia. Recognizing 3-d objects using surface descriptions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(11):1140–1157, 1989.
- [4] Feng Han and Song-Chun Zhu. Bottom-up/top-down image parsing with attribute grammar. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(1):59–73, 2009.
- [5] Chungan Lin and Ramakant Nevatia. Building detection and description from a single intensity image, 1998.
- [6] Brendan D. McKay. Practical graph isomorphism. In *Congressus Numerantium*, pages 45–87, 1981.
- [7] Y. Ohta, Takeo Kanade, and T. Sakai. An analysis system for scenes containing objects with substructures. In *Proceedings of the Fourth International Joint Conference on Pattern Recognition*, pages 752–754, 1978.
- [8] Yuichi Ohta. *Knowledge-based interpretation of outdoor natural color scenes*. Pitman Publishing, Inc., Marshfield, MA, USA, 1985.
- [9] A. Schförr. Programmed graph replacement systems. pages 479–546, 1997.
- [10] J. M. Siskind, Jr J. Sherman, I. Pollak, M. P. Harper, and C. A. Bouman. Spatial random tree grammars for modeling hierarchical structure in images with regions of arbitrary shape. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(9):1504–1519, 2007.
- [11] K. C. You and King-Sun Fu. A syntactic approach to shape recognition using attributed grammars. *Institute of Electrical and Electronic Engineers. Transactions on Systems, Man, and Cybernetics*, SMC-9(6):334–345, 1979.

POSTER: Segmentation Technique Based on Object Movement for Speech Production Simulation

Youhei Ohnishi

Osaka University
5-1 Mihogaoka Ibarakishi
567-0047 Osaka Japan

ohnishi.youhei@ais.cmc.osaka-u.ac.jp

Kazunori Nozaki

Osaka University
5-1 Mihogaoka Ibarakishi
567-0047 Osaka Japan

nozaki@ais.cmc.osaka-u.ac.jp

Ken-ichi Baba

Osaka University
5-1 Mihogaoka Ibarakishi
567-0047 Osaka Japan

baba@ais.cmc.osaka-u.ac.jp

ABSTRACT

In this paper, we propose the segmentation technique by using Snakes for 4dimensional magnetic resonance imaging (MRI) data of the movements of the oral tract shapes when pronunciation is performed. In the segmentation of the time-sequence MRI images, there is the specific problem that depends on a tongue shape quick deformation. We found that the Optical Flow of image sets is useful criteria for the decision of the geometry of control points in Snakes. Compared to the normal Snakes, our original method which modified Snakes by utilizing Optical Flow demonstrated superior accuracy of the segmentation of 4D MRI data.

Keywords

Segmentation, Snakes, 4dimensional MRI, Optical Flow

1. INTRODUCTION

For example, there are some studies of simulations for the blood flow by the heart beat and oral air flow by speech. In these simulations, the mesh for computation fluid dynamics is generated geometrical by using geometrical data. The geometrical data of the region of interest (ROI) is extracted image set such as MRI data. In these simulations, there are some temporal changes of ROI in time course. For example, the tongue moves when pronunciation is performed. Those organs movements should be taken into consideration. In order to represent movements of organs, we need to consider the changes of the geometrical data. These simulations often use Arbitrary Lagrangian Eulerian (ALE) method. In

ALE method, the changes of the geometrical data are represented by the shift and deformation of the mesh. In order to shift and deform the mesh, we need to get the velocity of the mesh nodes. An approach to get the velocity uses a mathematical model that represent the movement of geometrical data in time course [Watanabe04]. However, in the speech production simulation, it is impossible to use the mathematical model, because the movement of a tongue is volitional. So, the other approach to get the velocity of the mesh nodes without the mathematical model is required, which is to get the velocity from time-sequence image sets. In this approach, the geometrical data is extracted from image sets at each time step. And then, the difference of the geometrical data is computed between at time step T_n and T_{n+1} . It is computed how much each mesh nodes move between adjacent time step by using difference of the geometrical data. The information about movement of the mesh nodes means the velocity of mesh nodes.

Some segmentation methods of using Active Contour Model (ACM) have been proposed, which extract the geometrical data from time-sequence image sets. In our initial implementation, we segment 4D MRI data of oral tract shape while speaking by

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

using ACM simply. However, the problem is that the accuracy of segmentation depends on how much ROI changes between adjacent time steps. For example, if the tip of the tongue moves quickly between adjacent time steps, the result of segmentation tends to be unacceptable.

In this paper, we propose the segmentation technique by using Snakes with the specified criterion derived from Optical Flow in order to capture the quick motion of the tip of the tongue.

2. RELATED STUDY

Active Contour Model

Segmentation methods using ACM semi-automatically extract the geometrical data from time-sequence or spatial sequence image sets. When these segmentation methods are applied to an image in order to get the geometrical data of the boundary edge of ROI as a result contour (RC) of the segmentation, an initial contour (IC) is given on the image by the user and deformed based on some control rules such as some functions in order to close to the boundary edge of ROI. If these segmentation methods are used to segment time-sequence image sets, IC of the image at present time step is RC of the image at previous time step.

There are Snakes [Kass85] and Level Set Method (LSM) [Osher88] for two representative method using ACM. The main differences are how to represent active contour (AC) and control AC. In Snakes, AC is represented by control point (CP) set and deformed in a way that each CP is moved based on the evaluation function which is defined based on some features such as the shape of boundary edge of ROI. In LSM, AC is represented as the zero level set of an auxiliary function called the level set function (LSF), and deformed in a way that, the equation for the evolution of LSF is numerically solved.

The Difference between Snakes and LSM

There are two differences between Snakes and LSM. The first difference is that topology of AC can change only in LSM. The second difference is that Snakes uses the parametric representation for the AC, and LSM uses implicit function.

Optical Flow

Optical Flow is an approximation of the local image motion and specifies how much each pixel moves between adjacent images. We can get the information about velocity of object in the image by using Optical Flow. Generally speaking, Optical Flow computation methods can be classified into two categories: the dense Optical Flow method (the dense method) and

sparse Optical Flow method (the sparse method). Block matching method [Huang95] and Horn & Schunck algorithm [Horn81] are the dense method. In the dense method, the velocities of all pixels are calculated as Optical Flow. Lucas-Kanade (LK) method [Osher88] is also the dense method but it can be the sparse method. Pyramidal implementation of the LK (pyramidal LK) method [Bouguet00] is sparse method. In the sparse method, Optical Flow is computed only for an exclusive number of features. By computing Optical Flow only for some features, the accuracy of Optical Flow is able to be higher than the dense method. In order to obtain Optical Flow of the object which moves significantly, Pyramidal LK method is more effective than LK method.

3. PROPOSED METHOD

4D MRI Data of Oral Shape

The MRI data of oral tract shape in pronunciation of “/u/-/s/-/u/-/i/” has been taken by 3 tesla MRI system for a second, and covers before and after pronunciation. The partial resolution of the MRI data is 128*128 pixels and the temporal resolution is 16 images for a second. 7 slice images are taken for every time steps. So, there are 121 slice images in the MRI data.

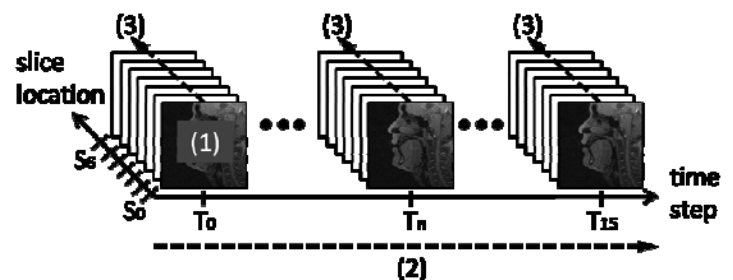


Fig.1: The segmentation process of 4D MRI data

Snakes to Segment 4D MRI Data

In the order to obtain the velocity of the mesh nodes from the difference of the geometrical data sets, there are two requirements for the segmentation to perform the simulations by using ALE method. The former one is keeping the same topology of the AC at every time step. The oral tract shape sometimes splits in two, develops holes, or the reverse of these operations by the motion of the tongue in pronunciation. In fact, the topology changes like these means that the oral tract gets thin or thick. In order to represent the movement of the tongue, the topology of AC must not change. The latter one is keeping the correspondence of the each point of the geometrical data between adjacent time steps. Because the geometrical data is used to compute the velocity of the mesh nodes. Those requirements

indicate that Snakes is superior to LSM for our purpose.

The 4D MRI data can be segmented by simply using Snakes. In simply using Snakes, the process of the segmentation of the MRI data consists of 3 phases ((1), (2) and (3) in Fig.1).

In the first phase ((1) in Fig.1), the user gives IC_{T_0,S_0} which is IC at time step T_0 and at slice location S_0 . IC_{T_0,S_0} is obtained in a way that Img_{T_0,S_0} which the image at T_0 and at S_0 is smoothed by the Gaussian filter and binarized by threshold.

In the second phase ((2) in Fig.1), the time-sequence images are segmented by using Snakes. RC_{T_0,S_0} is obtained by segmenting Img_{T_0,S_0} . Then, RC_{T_0,S_0} is used as IC_{T_1,S_0} , and Img_{T_1,S_0} is segmented. Thus, the segmentation of Img_{T_n,S_0} ($n \in 1 \sim 15$) is performed by using RC_{T_{n-1},S_0} as IC_{T_n,S_0} .

```

 $IC_{T,S}$  = IC used for segmentation of  $Img_{T,S}$ 
 $RC_{T,S}$  = the result contour of segmentation of  $Img_{T,S}$ 
for  $n = 1$ ..the number of time step do
     $RC_{n,S}$  = segment using Snakes( $Img_{n,S}$ ,  $IC_{n,S}$ )
     $IC_{n+1,S} = RC_{n,S}$ 

```

Algorithm 1: Pseudo-code of the segmentation algorithm in the second phase.

In the final phase ((3) in Fig.1) is the segmentation of the spatial-sequence images. The segmentation of Img_{T_n,S_1} is performed by using RC_{T_n,S_0} as IC_{T_n,S_1} . Similarly, the segmentation of Img_{T_n,S_k} ($k \in 1 \sim 6$) is performed by using RC_{T_n,S_k} as IC_{T_n,S_k} . Thus, 4 dimensional geometrical data is obtained by the segmentation of the temporal and spatial sequence images involved in the MRI data.

```

for  $n = 1$ ..the number of time step do
    for  $k = 1$ ..the number of spatial slice do
         $RC_{n,k}$  = segment using snake( $Img_{n,k}$ ,  $IC_{n,k}$ )
         $IC_{n,k+1} = RC_{n,k}$ 

```

Algorithm 2: Pseudo-code of the segmentation algorithm in the final phase.

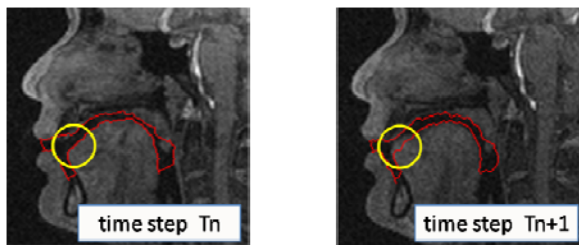


Fig.2: The problem depends on a tongue shape quick deformation.

There is the problem in the second phase of the method simply using Snakes. The problem is that the

accuracy of segmentation depends on how much ROI changes between adjacent time steps. For example, if the tip of the tongue moves significantly between adjacent time steps, the result of segmentation is unacceptable (the yellow circle in Fig.2). In order to solve this problem, the criterion of tongue movement incorporated into segmentation process.

Segmentation Technique Based on the Tongue Movement

In order to obtain more accurate RC by alleviating the above-mentioned problem which depends on a tongue shape quick deformation, it is effective to move IC closer to the boundary edge of ROI. In our proposed method, IC is refined based on the object movement estimated between adjacent time steps, and the movement is estimated by using Optical Flow. Now, we are interested in only ROI, not all pixels. So, the pyramidal LK method is utilized with IC as the features to calculate Optical Flow. Fig.3(a), and (b) show the shape of the tongue at time step T_n , T_{n+1} respectively. In these images, black line denotes the boundary edge of the tongue, and green line denotes the RC at time step T_n . In Fig. 3(a), red arrow means Optical Flow of all pixels on the RC calculated by pyramidal LK method. In our proposed method, the refined IC at the time step T_{n+1} is obtained in a way that the information of Optical Flow add to the coordinate data of RC at time step T_n . The refined IC is shown as a red broken line in Fig.3 (b).

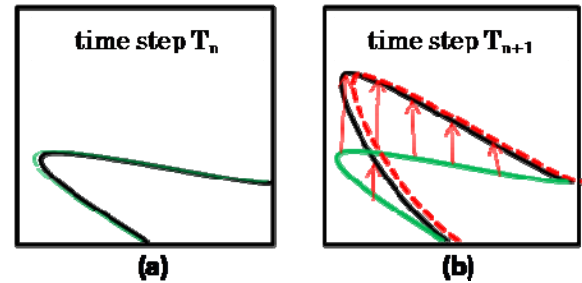


Fig.3: The shape image of tongue at time step of T_n , T_{n+1} respectively.

4. EXPERIMENTAL RESULT

To validate our proposed method, the method have been applied to 2 time-sequence images of the MRI data explained in Chapter 3. In these images, the tip of the tongue moves significantly. Fig.4 shows the results of this evaluation. Here, the window size is arbitrary in calculating Optical Flow using pyramidal LK method. The window of 13×13 pixels was suitable for estimation of the movement captured in the images used in this experimentation. Fig.4 (a) and (b) show respectively the picture at the time step T_n , and T_{n+1} , which are used in this experimentation. Fig.4 (c) is the picture extending the tip of the tongue

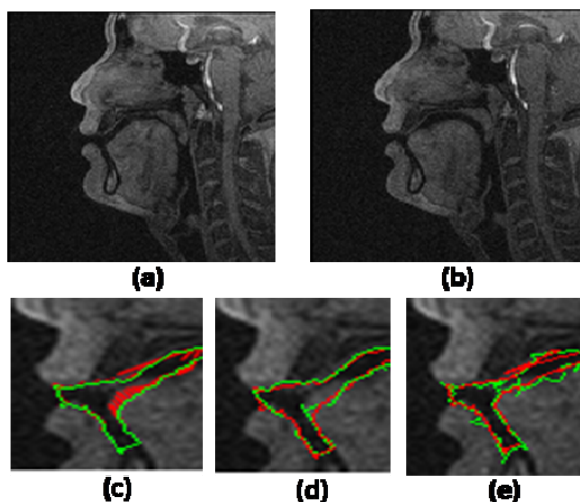


Fig4: The experimental result.

of Fig.4 (a). In Fig.4 (c), the green line and red line respectively show the RC at time step T_n and the Optical Flow calculated from these consecutive pictures. Fig.4 (d) and (e) are the pictures extending the tip of the tongue of Fig.4 (b). Fig.4 (d) shows the result of simply using Snakes and Fig.4 (e) shows the result of our proposed method. In Fig.4 (d) and (e), the red line plots the IC, and the green line plots RC. From these results, our proposed method is more effective than the method simply using Snakes, in order to extract the contour from the 2 time-sequence images in which the object in the images moves significantly.

5. CONCLUSION

In this paper, we proposed the segmentation technique by using Snakes for 4D MRI data of oral tract shape. Our proposed method overcame the oral tract shape deformation problem by utilizing the criterion of Optical Flow. Compared with the normal Snakes, the proposed method could improve the accuracy of segmentation. The proposed method is not taken into consideration of calculation cost so much, and so it is possible to be more rigorous segmentation by improving the accuracy of both the Snakes and Optical Flow.

6. FUTURE WORK

In our future work, we are going to segment all the oral shape MRI data by using our proposed method, and extract 4 dimensional geometrical data. In this

work, the window size has been determined experimentally according to the movement of interest. In order to determine the window size that is suitable for several movements, we should analyze more cases in which the method simply using Snakes can not be used to segment. Furthermore, we will generate the mesh from the 4 dimensional geometrical data, and simulate the oral air Flow by using the mesh. Then, we consider the availability of the mesh from the point of the calculation accuracy.

ACKNOWLEDGMENTS

We have the special thanks to National Institute of Information and Communications Technology (NICT) Project of "The development of the high-end device and applications connected with Broad band network", and Kobe Advanced ICT Research Center (KARC) to take the MRI data.

REFERENCES

- [Kass88] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Models, International Journal of Computer Vision, pp. 321-331, 1988
- [Lucas81] Lucas, B., and Kanade, T. An Iterative Image Registration Technique with an Application to Stereo Vision, Proc. of 7th International Joint Conference on Artificial Intelligence (IJCAI), pp. 674-679..
- [Bouguet00] Jean-Yves Bouguet. Pyramidal Implementation of the Lucas Kanade Feature Tracker.
- [Horn81] Berthold K.P. Horn and Brian G. Schunck. Determining Optical Flow. Artificial Intelligence, 17, pp. 185-203, 1981
- [Huang95] Y. Huang and X. H. Zhuang, "Motion-partitioned adaptive block matching for video compression", 1995.
- [Osher88] S.J. Osher, J.A. Sethian, Fronts propagating with curvature dependent speed. Algorithms based on Hamilton-Jacobi formulations. J. Comp. Physics, 79, 1988, 12-49, RESEARCH ARTICLE.
- [Watanabe04] Watanabe,H,Sugano,T, Sugiura,S Hisada,T, Finite Element Analysis of Ventricular Wall Motion and Intra-Ventricular Blood Flow in Heart with Myocardial Infarction. JSME Int J(C), 2004; pp 1019-1026.

Efficient acceleration structure layout for 64-bit many-core architectures

Maxim Shevtsov
Intel Corporation
30 Turgeneva Street,
603024, Russia, Nizhny Novgorod
maxim.y.shevtsov@intel.com

Alexei Soupikov
Intel Corporation
30 Turgeneva Street,
603024, Russia, Nizhny Novgorod
alexei.soupikov@intel.com



a) Asian Dragon model, 7.2M triangles, 64-bit extension consumes only 2Mb of 1.3Gb acceleration structure, extension processing time is <0.5% of rendering time

b) Thai Statue model, 10.2M triangles, 64-bit extension consumes only 2.1Mb of 1.4Gb acceleration structure, extension processing time is <0.5% of rendering time

d) Thai Statue model replicated 7 times 64-bit extension consumes only 4Mb of 7Gb acceleration structure, extension processing time is <0.5% of rendering time

Figure 1. 64-bit extension overheads for large (yet on-core) models. Models are ray traced with shadows at 1024x1024 on a 2-way 3GHz Intel ®Core™2 Duo machine (4 cores, 1 thread/core), 8Gb RAM, Vista64

ABSTRACT

A lot of rendering solutions use an acceleration structure to reduce the complexity of solving geometric proximity search problems. Although acceleration structures are well studied, data exceeding 32 bit address space require an acceleration structure with special properties, such as compact memory layout, efficient traversal capability, memory address space independence, parallel construction capability and 32/64 bit efficiency.

We propose a specific memory layout for a kd-tree and methods of processing that data structure handling massive models with the highest efficiency possible. The components of that are easily applied to other hierarchical acceleration structure types as well.

Keywords

Rendering, acceleration structure, kd-tree, ray- tracing, proximity search.

1. INTRODUCTION

Rasterization or ray tracing of models with large polygon counts usually rely on fast methods of geometrical proximity search. A good quality

acceleration structure reduces complexity of the search queries from $O(N)$ to $O(\log(N))$, where N is the number of primitives [Hav01]. The most efficient structures are based on non-balanced binary trees like kd-tree, BVH, BIH [WK06] or BSP, refer to [Hav01] for an overview. An acceleration structure practical for high-speed parallel processing must satisfy the following requirements:

- Efficient traversal capability – compact representation do not slow down the traversal step
- Memory address space independence –the acceleration structure is easy to save/load/transfer

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

- Parallel construction - the acceleration structure should support creation in multiple parallel threads
- 32 and 64 bit efficiency – the acceleration structure size should not explode on 64 bit architectures. The 32 bit mode acceleration structure mode should have exactly the same binary representation on 64 bit architectures.

In this paper we propose specific memory layout solving the above problems. We use kd-tree as example, but the solution we proposed is also applicable to a wide range of partitioning hierarchies. Furthermore it has a backward compatibility with previous layouts one may have.

2. PREVIOUS WORK

Kd-tree is a binary tree in which each node corresponds to a spatial cell. A kd-tree construction proceeds in a top-down fashion using a cost metric to determine split plane position in a current node until some termination criteria is reached and the node becomes a leaf. An inner node stores splitting plane position and references to the two child nodes. Each leaf node refers to a corresponding list of primitives.

The representation of a non-balanced kd-tree node requires a flag indicating whether the current node is an inner node or a leaf. The inner node stores a single address offset. Adding the offset to the memory address of a given node gives the memory address of the two child nodes [WBWS01]. A kd-tree node occupies eight bytes only. In combination with a proper memory alignment, the layout allows storing the split dimension in the two least significant bits of the offset. The highest bit indicates inner node or leaf, while the remaining 29 bits encode an unsigned address offset (substituting a pointer) to either the child nodes or to the list of primitive indices:

```
/* basic 8-byte layout for a kd-tree node */
struct KdTreeNode {
    union{
        //position of axis-aligned split plane
        float split_position;
        // or number of primitives in the leaf
        unsigned int items;
    }
    unsigned int dim_offset_flag;
    /*'dim_offset_flag' bits encode multiple data:
    // bits[0..1]: encode the split plane dimension
    // bits[2..30]: encode an unsigned address offset
    // bit[31]: encodes whether node is an inner node or a leaf
    */
    // macros for extracting node information
#define DIMENSION(n) (n->dim_offset_flag & 0x3)
#define ISLEAF(n) (n->dim_offset_flag & (UINT)(1<<31))
#define OFFSET(n) (n->dim_offset_flag & 0x7FFFFFFC)
```

Figure 2. Basic eight byte kd-tree node layout. Refer to [Ben06] or [Wal01] for details.

Storing offsets instead of pointers makes the data structure independent of its base address, thus no

pre-processing is required for storing/loading. The 31st (sign) bit of the offset field as leaf indicator results in efficient leaf/node test if offset are always non-negative. However, a 29 bit offset limits the displacement to 2^{29} bytes, which becomes insufficient for models larger than 10M triangles. Naïve replacement of 4 bytes with 8 bytes to get 61 bit offset on 64 bit machine leads to explosive growth of the memory footprint.

We propose to address that problem with a 64 bit extension mechanism that uses 32 bit offset field for the majority of nodes extending the offset to 64 bits for only a small fraction of nodes. Also a kd-tree fitting into 32-bit address space will have exactly the same binary layout on a 64 bit machine as it had on a 32-bit machine.

Positive offset ([Wal01]) assumes that child nodes are always located at higher addresses than their parents, which limits choices of memory allocation strategies, especially for multi-threaded builds. Construction threads usually allocate memory by continuous chunks. Once chunk is full a thread requests a new region from memory allocation system [SSK07]. A multi-threaded memory allocation system cannot guarantee positive offsets between branches of kd-tree constructed with multiple threads. So using positive offsets require additional transformation pass (similar to [ZHWG08]). We propose using negative offsets and that enables building using multiple regions rather than a single continuous array. To the best of our knowledge in-place construction of a kd-tree in multiple threads has never been done using offset-based representation.

As in [Wal01] we store both children of a node next to each other, both nodes are stored in the same cache line, so they're always fetched together automatically. However we noticed that less care was paid to the leaf/internal node test. As traversing a BSP node is by far the most frequent operation in a ray tracer, it has to be implemented with extreme care. Our leaf node test needs exactly one instruction before branch.

In a compiler field there is an intensive research on automatic pointers compression for 64-bit address space [LA05]. In our case only small number of nodes is really compressed/decompressed. As a result, even for high memory regions granularity the slowdown of rendering is less than 0.5% in compare to having 32-bit offsets only.

3. SOLUTION

A solution we propose still uses only eight bytes for kd-tree node layout. What is really new is how information is encoded and the amount of additional

information we manage to store within the same bytes, refer to Figure 3.

Since nodes and leaf data arrays are naturally aligned by 4-byte boundary, an offset between any two of them has 2 least significant bits available to store additional information:

- values 1, 2, 3 indicate internal node and split plane orientation (correspondingly X, Y, Z axis)
- value 0 is leaf indicator.

Our kd-tree node layout uses least significant bits for node/leaf flag, thus allowing for negative offsets. The changes in layout are highlighted in red below.

```

/* 8-byte layout for a kd-tree node */
struct KDTreeNode {
    union{
        float split_position;
        unsigned int items;
    }
    int dim_offset_flag;
    // 'dim_offset_flag' bits encode data in a new way
    // bits[0..1] : indicate either
    // • a leaf (if set to 0)
    //   // if 'items' field is >=0 it is true leaf
    //   // otherwise it is 64-bit extension
    // • an inner node with split plane dimension
    //   (if set to 1,2,3 for x,y,z axis corresp.)
    // bits[2..31] : encode a signed address offset
};

```

Figure 3. The proposed kd-tree node layout. Changes in layout are highlighted in red.

Efficient leaf/node test

During traversal the leaf/internal node test is executed at each traversal step, so its performance is critical. Having 0 as a leaf indicator (Fig.3) allows reducing the test to exactly 1 instruction before branch:

```

and    Node, 0x03
jz     ProcessLeaf

```

Our experiments with the proposed test demonstrated rendering performance improvement of ~5% on average (in compare to Fig.2 layout).

32 and 64 bit efficiency

To handle an unpredictability of a kd-tree size a construction algorithm allocates memory by reasonably sized continuous regions. The algorithm continues construction in the current region until it's full and then requests a new region from memory allocation system, Figure 4. As a result each such region contains a large connected portion of a constructed tree (one or more sub-trees). The number of links between those sub-trees is relatively small (<<1% of total number of links), thus the number of

nodes pointing to children located in another memory region is also small.

The typical region size is way smaller than 4GBs. So inside a continuous region the nodes can use 32-bit offsets as far as they reference children within the same region. The only nodes that potentially need 64-bit offsets are the nodes having children located in another memory region. *A node needing 64-bit offset is encoded as a special extension of a regular node.* To avoid frequent checks if a node is extended we extend leaves rather than internal nodes.

Multi-threaded construction

The tree is usually constructed in top-down manner from parent nodes to children nodes. When the tree is constructed in multiple threads each thread builds some sub-tree [SSK07]. Thus different threads may create a parent node and its children nodes. So when a parent is created the offset to children nodes may be unknown. That fact prevents from allocating 64-bit offset data next to a node (when 32-bit offset is insufficient). The actual data of 64-bit extended node is stored in a special per-thread relocation table:

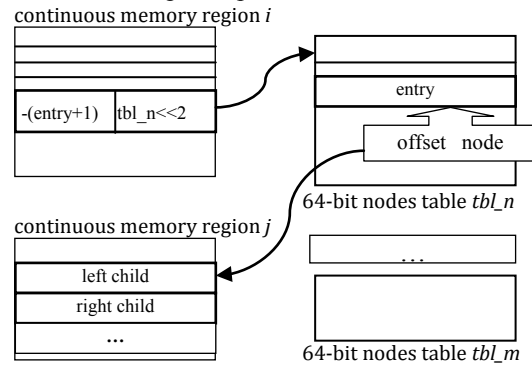


Figure 4. Mem. allocation by continuous chunks

As described in previous section, 64-bit extension node is a special type of leaf:

- a) $-(entry+1)$, where $entry$ is a table entry number, is stored in **items** field (see Figure 3). Negative value indicates special leaf. Adding 1 distinguishes from empty leaf;
- b) $(tbl) \ll 2$ where tbl is a per-thread table number, is stored in **dim_offset_flag** field. The shift is required to zero 2 least significant bits, indicating a leaf.

Each construction thread creates its own 64-bit node table. So there is no contention between threads for updating or reallocating (when full) the tables. Since each table is small its usage does not affect construction performance. Tests on models with up to 70M polygons demonstrated that 256-entry per-thread tables were never full. Storage or transmission of a tree located in multiple memory regions requires relocation of cross-region offsets. Since 64-bit node tables are exactly nodes with cross-region references,

the relocation operation is a simple update of nodes in the tables rather than a scan and update of the whole tree.

```

struct TableEntry{ // relocation table entries
    //actual leaf/node but with zero offset in dim_offset_flag
    KDTreeNode node;
    //true offset
    __int64 offset;
};
#define NOTLEAF(n) (n.dim_offset_flag&0x3)
#define DIMENSION(n) ((n.dim_offset_flag&0x3)-1)
#define IS_64BIT_EXT(n) (n.items<0)
#define MAKELEAF(n,its,ofs) n.items = its; \
    n.dim_offset_flag = ofs;

#define ENCODE64BIT_EXT(n,table_id,entry_id) \
    MAKELEAF(n,-(entry_id+1),table_id<<2)
#define DECODE64BIT_EXT(node, newadr) \
    int tab_id = (node.dim_offset_flag)>>2; \
    int entry_id = -node.items-1; \
    TableEntry e = m_tables[tab_id][entry_id]; \
    newadr = &node + e.offset; node = e.node;

```

Figure 5. Kd-tree layout+64-bit extensions macros

Modifications of traversal algorithm

The conventional 32-bit tree can be rendered by 64-bit code without any modifications. For the specific 64-bit extensions of the traversal algorithm, refer to Figure 6, Figure 7. Since the probability of traversing leaf is way smaller than probability of traversing internal node, the additional 64-bit extension test is performed at a very small fraction of traversal steps.

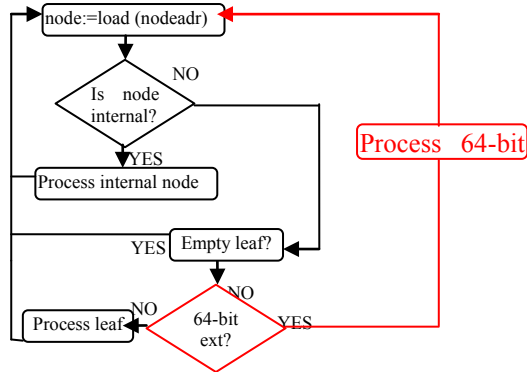


Figure 6. 64-bit extensions for traversal algorithm modification (in red).

4. Performance impact

Tests on large models demonstrated that with the proposed layout memory, footprint of trees constructed on 64-bit machine have almost the same size as the ones constructed on 32-bit machine (i.e. using 32-bit offsets only). Managing per-thread 64-bit node tables in our measurements demonstrated that construction slowdown is <1% and thus is negligible. We also performed tests for 2-128 construction threads with wide range of models (1-100M polygons). For all the tests, 64-entry per-thread tables are more than sufficient to connect portions of a tree constructed with different threads.

The performance of rendering using new layout supporting 64-bit extensions is the same as of rendering the efficient layout supporting 32-bits only (see Figure 1 for examples). Even on complex models and high memory region granularity the slowdown using the proposed layout was less than 0.5% comparing with 32-bit only offsets and one continuous memory region for the whole tree.

```

register KDTreeNode node = m_root;
// ADRINT is int or __int64 (32/64-bit architectures)
ADRINT newadr = &node;
traverse_loop:
while (NOTLEAF(node)){
    //get dimension, traversal order, etc
    const ADRINT adr0 = newadr+...; //front child
    const ADRINT adr1 = newadr+...; //back child
    //traverse of either back/front child or both
    ...
}
//processing leaves
if (node.items > 0){
    ...
}
}
#ifdef _M_X64
else if (IS_64BIT_EXT(node)){
    //64-bit extensions processing:
    //newadr is patched using relocation table
    DECODE64BIT_EXT(node, newadr);
    goto traverse_loop; //another option is to duplicate
    traversal/leaf-processing code here
}
#endif // _M_X64

```

Figure 7. Pseudo-code for handling of 64-bit extensions in the traversal algorithm (in italic/bold)

5. Future Work

Transparent support of multiple continuous memory regions that we proposed, allows implementing simple and efficient paging/caching mechanisms in spirit of [YM06].

6. REFERENCES

- [Ben06] C. Benthin, "Realtime Ray Tracing on current CPU Architectures", PhD thesis, Saarland University, 2006.
- [Hav01] V. Havran: "Heuristic Ray Shooting Algorithms". PhD thesis, Czech Technical University in Prague, 2001.
- [LA05] C. Lattner and V. Adve: "Transparent Pointer Compression for Linked Data Structures". In Proceedings of the ACM Workshop on Memory System Performance (2005).
- [SSK07] M. Shevtsov, A. Soupikov, and A. Kapustin.: "Highly parallel fast kd-tree construction for interactive ray tracing of dynamic scenes". In Proceedings of Eurographics (2007).
- [Wal01] I. Wald, "Realtime ray tracing and interactive global illumination", PhD thesis, Saarland University, 2004.
- [WBWS01] I. Wald, C. Benthin, M. Wagner, and P. Slusallek, "Interactive Rendering with Coherent Ray Tracing". Computer Graphics Forum, 20(3) (2001).
- [WK06] C. Wächter, A. Keller.: "Instant Ray Tracing: The Bounding Interval Hierarchy". In Proceedings of 17th Eurographics Symposium on Rendering (2006).
- [YM06] S.-E. Yoon, D. Manocha: "Cache-Efficient Layouts of Bounding Volume Hierarchies". Computer Graphics Forum 25(3) (2006).
- [ZHWG08] Kun Zhou, Qiming Hou, Rui Wang, Baining Guo: "Real-time KD-tree construction on graphics hardware" In Proceedings of ACM SIGGRAPH Asia (2008).

High Resolution 3-D Face Modeling and Model Warping

Ersin Özüağ

M. Kemal Güllü

Oğuzhan Urhan

Sarp Ertürk

Kocaeli University Laboratory of Image and Signal Processing (KULIS)

Electronics & Telecom Eng. Dept. Umuttepe Campus, 41380, Kocaeli, TURKEY

ersinsys1@gmail.com kemalg@kocaeli.edu.tr urhano@kocaeli.edu.tr sertur@kocaeli.edu.tr

ABSTRACT

In this work, 3-D representations of human faces are obtained at a computer by making use of a DLP projector and a high resolution camera. The system calibration is carried out using a calibration pattern image. A colored structured light approach is adopted in this paper instead of the traditional gray-tone pattern to obtain depth information. A heuristic color correction approach is proposed in this work to improve the performance of the previous approaches. Experiments show that high resolution depth information can be extracted with this approach. Furthermore, a GUI is designed to enable user controlled modifications of the 3-D model.

Keywords

Computer vision, Structured light, 3-D Modeling, Model Warping.

1. INTRODUCTION

3-D (3-dimensional) technologies have applications in medicine, biology, unmanned control systems, and object recognition. An easy way to obtain 3-D representations of real world objects is to use structured and coded light [1].

Laser strips can be used with a rotating mechanism to create 3-D representation of objects [2]. Another way is to form a structured light using a projector and capture this pattern using a camera. The advantage of the later approach is that it does not require any rotating mechanism to obtain the 3-D model. Thus, compact and low-cost devices that are able to create 3-D models can be designed using this approach.

In this work, a low-cost and efficient 3-D modeling and modification system is designed similar to [3] with an enhanced color correction approach novel to this work. Furthermore, a graphical user interface is designed to enable user to make modifications on the model obtained in the computer.

2. 3-D MODELLING SYSTEM

The main idea of the 3-D modeling system used in this work is to follow changes in the reflections of the projected structured light. This idea is depicted in Fig. 1(a).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

2.1. Hardware

The system used for 3-D model creation is composed of three main components: projector, camera and computer. A typical placement of these components is shown in Fig 1(b). The system is controlled by the computer. Our setup consists of a projector that has a resolution of 1024x768 pixels, 3000 ANSI lumens of brightness, a camera which is capable of capturing images at 12Mpixel and notebook computer which controls the whole system.

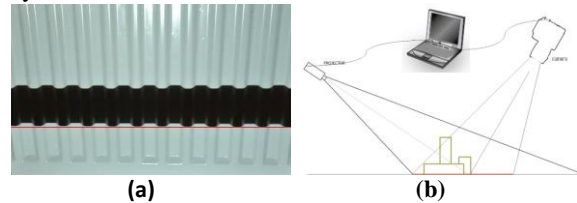


Fig.1: (a) 3-D scanning using structured light approach, (b) A typical placement of components used in the 3-D scanning system.

2.2. 3-D Model Creation Method

Calibration plays an important role to find the relation between the real world and 3-D world created on computer [4]. The calibration process can also compensate for errors originated from the optics.

Since a high resolution camera is utilized to capture images the method used for the extraction of the depth information has to be computationally efficient to process the high amount of data. Thus, a special structured light pattern has to be chosen so that less computation will be required for separation of the strips. The pattern used in this work is shown in Fig 2(a) [3]. Fig. 2(b-c) shows a raw test image

and its structured light projected form.

After obtaining the image as in Fig 2(c) the next step is to determine the face area. We used user supervision at this stage for simplicity. General framework of the proposed 3-D modeling approach is depicted in Fig. 3(a) [3]. The image obtained after the face selection process is given in Fig. 3(b).

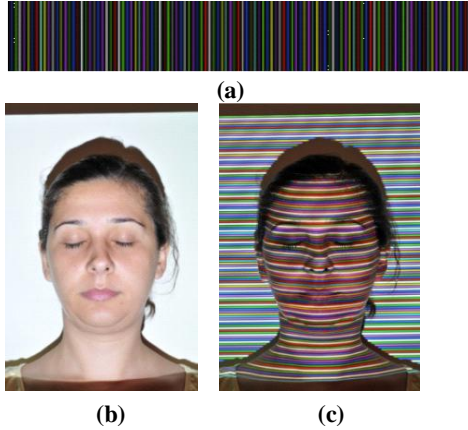
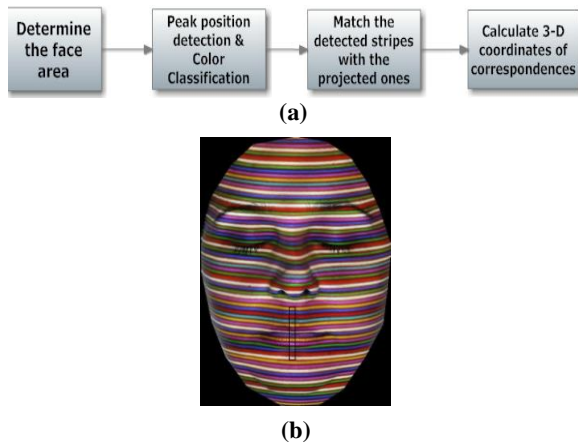


Fig. 2: (a) Color pattern used for 3-D model creation (rotated by 90°) [3], (b) Original image (used for texture information), (c) Structured light projected form of (b).



**Fig. 3: (a) Flowchart of the proposed method [3]
(b) Image obtained after face region selection process.**

A 2-D hourglass shaped low-pass filter and 1-D band-pass filter is employed successively to remove noise and facilitate easier tracking of the strips as in [3]. Note that the filtering operations not only remove the noise but also provide a more convenient image for the separation of the color strips.

The effect of filtering operations on the color peak values are shown in Fig. 4. The figure shows the color value changes in terms of RGB values for the highlighted area in Fig. 3(b). After enhancing the input image the next step is to detect color peaks in the filtered image to follow the strips. We propose to

find the peak values in a heuristic way instead of polynomial fitting used in [3].

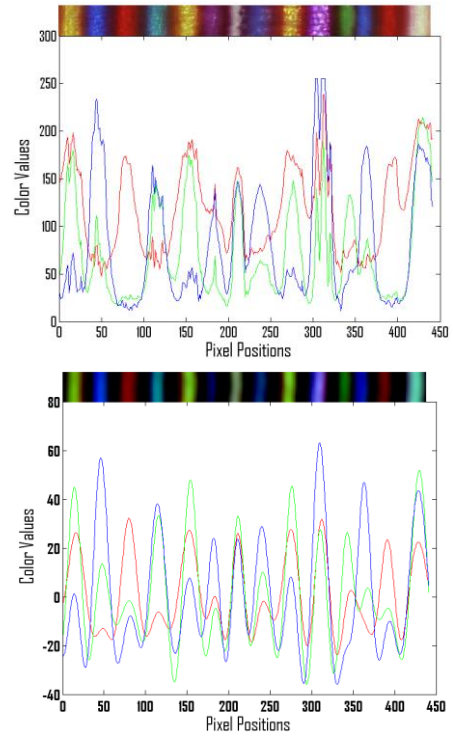


Fig. 4: The effect of filtering operations on the color peak values. Before filtering (up), After filtering (down).

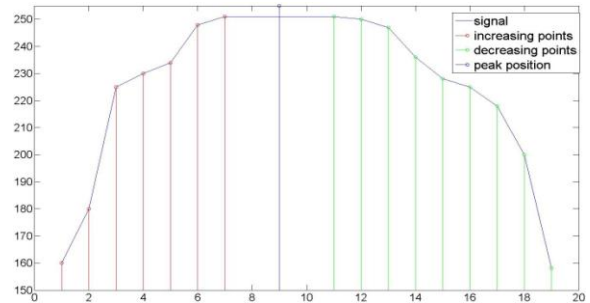


Fig. 5: Heuristic peak position determination.

Fig. 5 shows how our heuristic approach works. It firstly follows the changes of the peak values along the strip and then detects two critical points at which increase in the amplitude finishes and decrease starts.

Peak value positions are computed in a similar way for each color channel. Next, peak positions for each channel are averaged to find vertical peak color positions as in [3]. There are some misclassifications at this stage which might significantly affect quality of the final model. In [3], a k-means based line fitting method (KMeansLineFit) is utilized to eliminate misclassifications examining distributions of pixel values in RGB space (Fig. 6(a)). This method fits seven lines to the RGB space and decides the class of the pixel according to the distance between the line and the point.

Our experience has shown that when the amplitude values of pixels are below a certain value, the classification performance might be degraded. In order to improve the performance we propose to use a weighting approach before the classification stage as given below:

$$I_w = I + I_f \times w \quad (1)$$

where (1), I , I_f , I_w , and w represent the original image, filtered image, enhanced image and weighting coefficient, respectively.

The pixel value distribution in RGB color space after the weighting operation is shown in Fig. 6 (b). As seen from this figure, distinction between the classes is more obvious. The enhanced part of the highlighted image area in Fig. 3(b) is also shown in Fig. 6 (c). The KMeansLineFit method is executed after the weighting process.

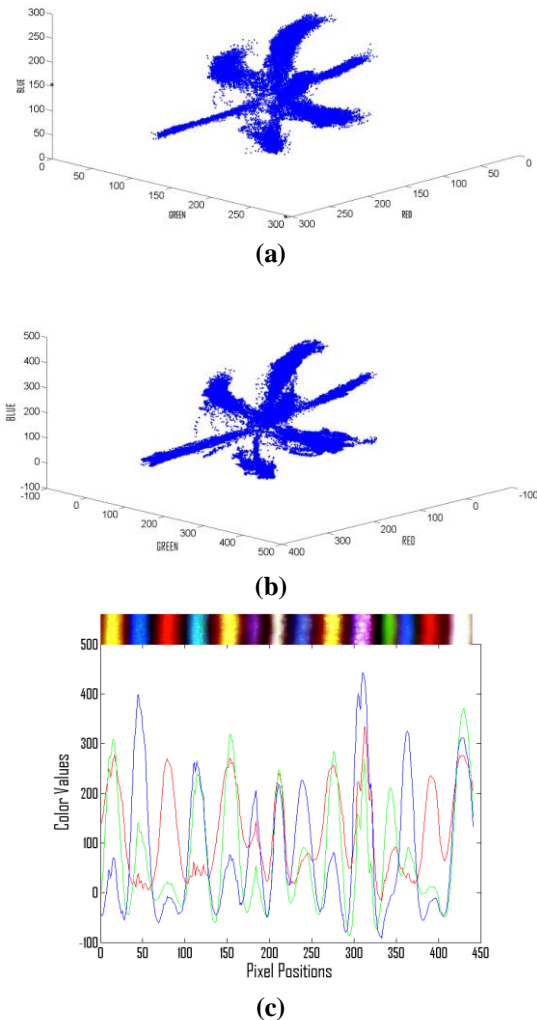


Fig. 6: (a) RGB space distribution of the pixels that corresponds to peaks in Fig 3(b). (b) Same distribution after the weighting (c) Color values after the enhancement

Although the weighting operation reduces the number of color strip misclassification, there are still some incorrect color assignments. If there is more than one color in a strip then majority voting is carried out within the strip and a fixed color is assigned to all pixels belonging to that strip. Next, strip segments that do not have any connection with other segments are examined by taking the reference strip into consideration and gaps between the segments are linearly fitted. An original image part and the corresponding color classification obtained at this step are shown in Fig. 7.

The position differences between the originally projected and observed lines are required to compute the depth. For this purpose, the reference lines that are obtained using camera-projection calibration [5] are compared to strips obtained as in Fig. 7 to extract the depth information. Depth information is depicted as 3-D form in Fig. 8.

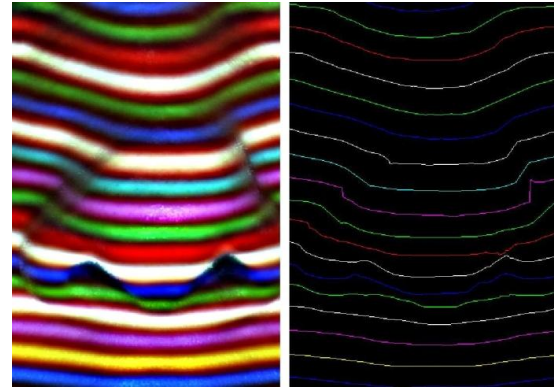


Fig. 7: Example enhanced image part and corresponding color classification of the proposed approach.

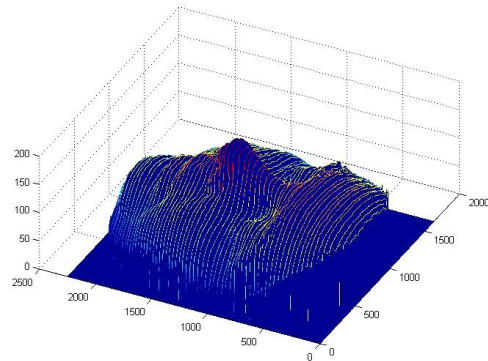


Fig 8: Representation of the extracted depth data

After the depth information is obtained the data has to be triangulated for further processing. In this work the Delaunay triangulation method [6] is used for this purpose.

2.3. 3-D Model Warping

After the triangulation process, the data can be visualized in different forms. Example visualization with texture mapping is given in Fig 9. OpenGL

software interface is commonly used in 3D applications [7-8].

One of the main purposes of this work is to make modifications on the 3-D model by means of a GUI. The modification on the model corresponds to changing some of the triangle coordinates of the model in 3D space. After the selection of the deformation area, deformation type and effect region must be defined. In our application, deformation region shape is chosen as a disk but it could be any smooth geometric shape. The effect of the deformation should decrease by distance to the deformation center. This effect is carried out by introducing a weighting coefficient.

$$X2 = X1 + k * (1 - dist / R) \quad (2)$$

$$X2 = X1 - k / (1 - dist / R) \quad (3)$$

In (2) and (3), $X1$, $X2$, k , and $dist$ show old and new positions of the vertex, the weighting coefficient and distance to the deformation center, respectively. The function given in (2) is used for expansion whereas the function in (3) is used for diminishing of the area. Fig. 9 shows original models (left column) and deformed model (right column) using the function in (2) and (3). Note that for example, in Fig. 9, nose of the female model is bigger than the original model.

3. CONCLUSIONS

We present a modified method for 3-D face model creation in this work. The method is based on the structured light approach and it work well for face modeling. Enhanced color correction approach presented in this work improves modeling accuracy. We also designed a graphical user interface for model modifications. Experimental results show the effectiveness of the proposed method. Additional modification operators will be considered for further work.

4. ACKNOWLEDGEMENT

This work was supported by Ministry of Industry, Trading of Turkey and Pars AR-GE Inc. under contract, 00262.STZ.2008-1.

5. REFERENCES

- [1] S. Winkelbach and F. M. Wahl. Shape from Single Stripe Pattern Illumination. Pattern Recognition, Lecture Notes in Computer Science 2449, Springer 2002, page 240-247.
- [2] İ. Yıldırım, B. Demir, S. Ertürk. 3-Dimensional Modelling and Visualization of Cultural Assets For E-Government. Signal Processing and Communications Applications, 2006 IEEE 14th, On pages:1-4.

- [3] P. Fechteler and P. Eisert. Adaptive Colour Classification for Structured Light Systems. IET Comput. Vis., 2009, Vol.3, Iss. 2, pp. 49-59
- [4] F. Sadlo, T. Weyrich, R. Peikert, M. Gross. A Practical Structured Light Acquisition System for Point-Based Geometry and Texture. Proceedings of the Eurographics Symposium on Point-Based Graphics 2005, Page(s):89-145
- [5] M. Ashdown, R. Sukthankar. Robust Calibration of Camera-Projector System for Multi-Planar Recognition Displays. HPL – 2003-24
- [6] en.wikipedia.org/wiki/Delaunay_triangulation (2009 -January)
- [7] OpenGL Programming Guide. Fifth Edition
- [8] M. Segal and K.Akeley. The OpenGL Graphics System. A Specification (Version 2.1 - July 30, 2006)



Fig. 9: Original models (left column) and deformed models (right column)

POSTER: Augmentation of Mean-Shift Method to Facilitate Face Tracking

Albert Akhriev

IBM Dublin Software Lab

Building 6, IBM Technology Campus, Damastown Industrial Park, Mulhuddart, Dublin 15, Ireland

aaahaaah@hotmail.com

ABSTRACT

In this paper we introduce an improvement to the well known mean-shift color tracker. On each new frame we estimate the histogram resolution which provides the best separation between color distributions of the object and the background. The optimal resolution is derived from the principle of minimum uncertainty of foreground/background classification. The augmented mean-shift method with variable histogram resolution was applied to the task of face tracking.

Keywords

Mean-shift color tracker, histogram resolution, Renyi entropy.

1. INTRODUCTION

Creating interactive computer systems that are effective and easy to use is an important direction of modern research. For example, face and hand tracking software can be a part of a human-computer interaction system that creates the illusion of feedback between user and an OpenGL application running on a personal computer. In this paper we focus on the *face tracking* method based on the well known mean-shift color tracker [Com03].

Recent developments have shown the effectiveness of color based tracking algorithms for objects with variable appearance. These methods can be separated into geometric (contour, region), feature-based (color, texture) and hybrid ones. Feature-based and hybrid methods are of particular interest in recent years and can be further subdivided into distribution tracking [Com03, Goo02, McK99, Zha05] and feature classification [Ngu02, Bra98] methods.

Distribution tracking is a relatively robust process, which locks firmly on an object's sub-region over multiple frames; however, it might not follow the shape and orientation changes.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

A typical feature classification method uses a model (e.g. histogram) to classify points of a new frame into object or non-object ones. Usually the model is deduced from past observations. The result of a classification is a 2D-array of weights or probabilities, where adjacent points with high values are aggregated together to form an object. This approach is geometrically flexible, but also error prone since it can produce false positives. If a distribution is dispersed across a feature space (e.g. histogram bins are not densely populated), a classification might be unreliable. Fortunately, face color distribution is usually compact.

We used the mean-shift tracker [Com03], which has been proved to be robust and computationally effective, as a reference. In order to reduce the sensitivity of color tracking methods to the presence of object-like features in the background, we came up with the following contributions:

- 1) We replaced the ad-hoc background utilization proposed in [Com03]. Instead, the histogram resolution, which provides the optimal separation between foreground and background, is computed in each new frame [Akh07].
- 2) Given estimation of object's location in a new frame, obtained by the mean-shift tracker, we generate a *probability map* (see Figure 2), where each pixel keeps the probability of object presence. Our aggregation method uses the probability map to make the final estimation of object's position.

2. COLOR TRACKING ALGORITHM

Optimal Histogram Resolution

Hereinafter we talk about color features only and color histograms as the distribution models. We assume that the reader is aware of the basic concept of the mean-shift tracking approach [Com03].

One possible way to find the optimal histogram resolution is to choose the one that minimizes statistical distance between foreground and background color distributions in the current frame, where object location is known. The best resolution found is used by the tracker in the next frame. The drawback of this approach occurs from unstable lighting conditions and low quality video output from cheap web-cameras. In contrast, our method takes into consideration a portion of the *next* frame, which likely contains the object being tracked.

Let us begin with some notation. At each moment t on the frame I_t the face occupies an *elliptic* region \mathbf{r}_t obtained as the result of previous calculations. At the moment $t = 0$ initial region is given. Let us introduce the set of face or *foreground* points F_t and the set of *background* points B_t . These sets form two regions inside and outside of face respectively (left picture on Figure 1). The set $C_t = F_t + B_t$ includes all points of the *current* frame that participate in the tracking process. Let us define the set of points N_t of the *next* frame I_{t+1} covered by the set C_t shifted on the vector of mean interframe displacement $T_t = (dx, dy)$ (right picture on Figure 1). We do not know the exact face position at the moment $t+1$, but we assume that it lies somewhere inside N_t . A color vector $\mathbf{c} = (R, G, B)$ is given at each image point. Color histograms H_f , H_b and H_n are constructed on the point sets F_t , B_t and N_t respectively. Let us denote a histogram entry for a color \mathbf{c} as $H(\mathbf{c})$, and the number of samples as $|H|$.

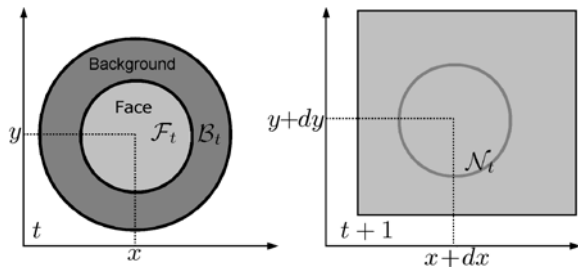


Figure 1. Regions on the current frame (left) and candidate region on the next frame (right).

The Idea of Optimal Histogram Synthesis

A color component R , G or B varies within the interval $[0, 255]$. We define *histogram resolution* as the number of bins this interval is divided into. Optimal resolutions of channels R_{bin} , G_{bin} and B_{bin} are obtained by minimization of classification uncertainty

$$U = \min_{R_{bin}, G_{bin}, B_{bin}} \sum_{\mathbf{c} \in N_t} H_n(\mathbf{c}) E(H_b(\mathbf{c}), H_f(\mathbf{c})), \quad (1)$$

where $H_n(\mathbf{c})$ is the number of points of the set N_t with color \mathbf{c} , and $E(\cdot)$ is the entropy (uncertainty measure) of classification of a point with color \mathbf{c} . The entropy function receives the numbers of samples with color \mathbf{c} in the background and foreground distributions as its arguments. Summing in (1) is carried out over all non-zero entries of the histogram H_n .

The idea of (1) is to minimize classification uncertainty over a region of the next frame, which contains an object being tracked, using foreground and background histograms accumulated so far at the current frame. We assume that histograms of the best resolution separate foreground and background points on the *next* frame with the least uncertainty. This is not necessarily true, but the heuristic works well in practice.

In order to reduce the mathematical complexity, we choose equally populated sets B_t and F_t ($|B_t| = |F_t|$) by proper selection of the background ellipse (the outer ellipse on Figure 1) or downscale the largest histogram.

Minimization of (1) is achieved by direct enumeration of histogram resolutions R_{bin} , G_{bin} and B_{bin} . We divide the interval $[0, 255]$ into 4, 8, 16 and 32 bins. Three channels give $4^3 = 64$ partitions of color space. The best partition minimizes (1).

Entropy

In this paper, *point classification* means assignment of a background p_b and a foreground p_f probabilities to a point with color \mathbf{c} on the next frame. Renyi entropy of such classification is given as follows:

$$E(\mathbf{c}) = -\ln(p_b^2(\mathbf{c}) + p_f^2(\mathbf{c})), \quad (2)$$

where empirical probabilities can be defined, for a while, via frequency of occurrence in histograms H_b and H_f respectively, e.g. $p_f = H_f(\mathbf{c}) / (H_b(\mathbf{c}) + H_f(\mathbf{c}))$. Both Shannon and Renyi entropies produce similar results, but there exists efficient approximation to Renyi entropy: $E = \text{const} (p_b p_f + 2.15544 (p_b p_f)^2)$.

In practice formula (2) does not work correctly because many histogram bins are populated with only few samples. In [Akh07] we have shown that more realistic results can be obtained by averaging the entropy over an unknown “true” state. The important consequence is that the average entropy significantly increases, comparing with the theoretical one (2), as the number of samples in a histogram bin becomes too small. *The optimal histogram resolution compromises between good separability of background and foreground distributions (high resolution) and low classification uncertainty (low resolution, many samples in a bin).*

The first term in Renyi entropy approximation gives the following expression after averaging (see [Akh07])

$$\bar{E}(\mathbf{c}) \approx \frac{\text{const} \cdot (1+H_b(\mathbf{c}))(1+H_f(\mathbf{c}))}{(2+H_b(\mathbf{c})+H_f(\mathbf{c}))(3+H_b(\mathbf{c})+H_f(\mathbf{c}))}.$$

We call it the *discrete entropy*. Discrete entropy participates in minimization (1). Then the optimal histogram resolution is used to run the mean-shift tracker. Similarly we calculate expectation of empirical probability, which takes into account a (potentially) small number of samples in a histogram bin, [Akh07]

$$\bar{p}_f(\mathbf{c}) = (1 + H_f(\mathbf{c})) / (2 + H_b(\mathbf{c}) + H_f(\mathbf{c})). \quad (3)$$

Blob Aggregation

The original mean-shift tracker [Com03] does not support plane rotation of an object. On the other hand, the cam-shift method [Bra98] provides flexible adaptation of an elliptic face, but often fails when the subtle balance between optimal window size and actual space distribution of skin-like points is broken. We have found that ideas from both methods can be unified into a better approach:

1. Given parameters of a face ellipse on the current frame I_t (center position, large and small semi-axes, angle between x axis and large semi-axis), find the optimal histogram resolution.
2. Compute foreground (H_f) and background (H_b) histograms with optimal resolution. Feed the histogram H_f into the mean-shift tracker.
3. Run the mean-shift tracker 9 times with scale factors $\{0.9, 1.0, 1.1\}$ in the directions of ellipse axes and pick up the best estimation [Com03].
4. Keeping the same value of ellipse area, change estimated ellipse so that the ratio between the large and the small semi-axes does not exceed 1.2 [Bir98, Bra98]. The latter improves the overall stability of the color face tracker.
5. Take in turn all points of the set N_t and calculate foreground probability (3) at each point. Points outside N_t receive default probabilities 0.5. As a result we obtain a *probability map* similar to back-projected image in [Bra98], see Figure 2. Every entry of the map keeps the face presence probability at a point of the next frame I_{t+1} .

The final steps correct the orientation and semi-axes of the face ellipse in the next frame:

1. Scale the ellipse, estimated so far, by a factor from the set $\{0.90, 0.92, \dots, 1.0, \dots, 1.08, 1.10\}$.
2. Compute covariance matrix \mathbf{C} over the *area of* each *scaled* ellipse, $\mathbf{C} = ((M_{xx}, M_{xy}), (M_{xy}, M_{yy}))$,

$$M_{uv} = \frac{\sum_i p(x_i, y_i) k(x_i, y_i) (u_i - u_c)(v_i - v_c)}{\sum_i p(x_i, y_i) k(x_i, y_i)}$$

where $u, v = \{x, y\}$, (x_i, y_i) is a point inside a scaled ellipse, (x_c, y_c) is an ellipse center, $p(x_i, y_i)$ is the value of probability map entry (3) and $k(x_i, y_i)$ is a Epanechnikov kernel employed in [Com03] that falls down to zero at the elliptic boundary of the face (we use the similar kernel in the main algorithm).

3. Estimate new ellipse parameters as it was done in [Bra98]. To prevent new ellipse from shrinkage, we have to “normalize” its semi-axes (next step).
4. Compute another covariance matrix \mathbf{C}_1 assuming $p(x_i, y_i) = 1$. Uniform probability map means no change in ellipse geometry should be done and the product of eigen-values of \mathbf{C}_1 must be equal to the squared product of ellipse’s semi-axes a and b (which are eigen-values in fact): $s^2 \det(\mathbf{C}_1) = a^2 b^2$, where s is a normalization factor we need to accomplish the step 3.
5. For each scaled and “normalized” ellipse compute the mean foreground probability $\langle p_f \rangle$ inside face area F_t and the mean background probability $\langle p_b \rangle$ inside background area B_t , see Figure 1. The quality (probability) of corrected estimation is defined as follows:

$$P = \sqrt{\langle p_b \rangle_{B_t} \langle p_f \rangle_{F_t}} = \sqrt{(1 - \langle p_f \rangle_{B_t}) \langle p_f \rangle_{F_t}}.$$

6. Pick up the scaled and “normalized” ellipse with the highest probability P .

Let us now summarize the most important points:

- 1) In contrast to [Bra98], we do not query points outside the (scaled) face area while computing moments M_{uv} . This protects us from outliers.
- 2) We trust the center position found by the mean-shift tracker, but we try to adjust the ellipse rotation and semi-axes on each new frame by direct enumeration of scale factors ($[0.9 \dots 1.1]$).
- 3) The normalization step is an important mechanism that stabilizes the ellipse nearby the most prominent blob in the probability map.
- 4) At the optimal ellipse location, the contrast between the face and the background regions on the probability map attains maximum.
- 5) Average probabilities and moments can be computed as the contour integrals, if we prepare an integral image [Vio04] from the probability map. This provides tremendous performance gain.

3. EXPERIMENTS

During experiments we used a standalone implementation of the color tracker. The face detector, proposed in [Vio04] and implemented in the OpenCV library, was invoked on the first frame to initialize the face position.

Currently we have an efficient C++ implementation of the module that computes the optimal histogram resolution and creates the probability map (about 2 ms per frame), a relatively efficient mean-shift tracker implementation (6-7 ms with 9 scale combinations) and a blob aggregator (2.5-3 ms). The typical running time is 10-12 ms per frame on Intel Pentium 2.6 GHz, 2 Gb, single thread. We are aiming to double the performance in the next version.

Our experiments confirmed that 16x16x16 histograms adopted in [Com03] have optimal resolution most of the time, see Table 1. However, even a few “difficult” cases might be crucial for the overall stability of the method, so the resolution optimization is useful.



Figure 2. The typical probability map. Each pixel keeps the probability of object presence.

The paper describes the ongoing project, though some observations could be summarized at this stage:

1. The augmented mean-shift tracker is stable, it tolerates moderate occlusion, including occlusion by the objects of the same color (e.g. hands), and it is able to recover by itself in many situations.
2. Methods based on skin detection (e.g. [Bra98]) may fail to start, if the camera settings are not appropriate, whereas our feature classifier almost always produces a dense blob.
3. We have noticed that slow face histogram updating, adopted in [Goo02], distorts rather than improves results. This is due to changing of lighting conditions and camera instability.
4. Some particular situations cannot be handled by a color tracker only. Open chest, neck or shoulders are unavoidably detected along with the face; see Figure 2, causing a drift and loss of

face ellipse. The problem could be solved by employing geometric methods, e.g. [Bir98], in addition to the color tracking approach.

5. A color tracker should be considered as an auxiliary tool that quickly localizes a face in a new frame. The next step after localization should include face detection (within a limited area), or deformable template matching, e.g. [Vio04, Dor06].

	4 bins	8 bins	16 bins	32 bins
Red	4.20%	40.4%	54.5%	0.89%
Green	2.29%	2.42%	50.2%	45.1%
Blue	0.13%	1.91%	76.8%	21.1%
All	2.21%	14.9%	60.5%	22.4%

Table 1. A typical bin number distribution of the optimal histogram per color channel.

4. REFERENCES

- [Bir98] Birchfield, S. Elliptical Head Tracking Using Intensity Gradients and Color Histograms. Proc. of the IEEE Conf. on Comp. Vis. and Patt. Recog., St Barbara, California, pp.232-237, 1998.
- [Bra98] Bradski, G. R. Computer vision face tracking as a component of a perceptual user interface. In Workshop on Applications of Comp. Vision, Princeton, NJ, 214-219, 1998.
- [Com03] Comaniciu, D. Ramesh, V. Meer, P. Kernel-based object tracking. IEEE Trans. PAMI 25(5), pp.564-577, 2003.
- [Dor06] Dornaika, F, Ahlberg, J. Model-based head and facial motion tracking. ECCV workshop on HCI, pp.221-232, 2004.
- [Goo02] Nummiaro, K., Koller-Meier, E., Van Gool, L. Object Tracking with an Adaptive Color-Based Particle Filter. In Proc. Symp. for Patt. Recog. of the DAGM, pp.353-360, 2002.
- [McK99] McKenna, S. J., Raja, Y., Gong, S. Tracking colour objects using adaptive mixture models. Image and Vis. Comp. 17(3-4), pp.225-231, 1999.
- [Ngu02] Nguyen, H.T., Worring, M., Boomgaard, R., Smeulders, A. W. M. Tracking nonparameterized object contours in video. IEEE Trans. Image Processing 11(9), pp.1081-1091, 2002.
- [Vio04] Viola, P., Jones, M. J. Robust Real-Time Face Detection. International Journal of Computer Vision 57 (2), 137-154, 2004.
- [Zha05] Zhang, T., Freedman, D. Improving Performance of Distribution Tracking through Background Mismatch. IEEE Trans. PAMI 27(2), pp.282-287, 2005.
- [Akh07] Akhriev, A. Object Tracking Via Uncertainty Minimization. ISVC(2), pp.592-601, 2007.

POSTER: Point Cloud Lossless Compression

Koji Nishio

Osaka Institute of Technology
1-79-1, Kitayama
Hirakata, Osaka
Japan(573-0196)
nishio@is.oit.ac.jp

Yusuke Takebayashi

Osaka Institute of Technology
1-79-1, Kitayama
Hirakata, Osaka
Japan(573-0196)
takebayasi@is.oit.ac.jp

Yuji Teshima

Shizuoka Institute of Science and
Technology
2200-2 Toyosawa
Fukuroi, Shizuoka
Japan(437-8555)
teshima@cs.sist.ac.jp

Takayuki Kanaya

Hiroshima International Univ.
555-36, Kurose Gakuendai
Higashi Hiroshima, Hiroshima
Japan (739-2695)
t-kanaya@hw.hirokoku-u.ac.jp

Ken-ichi Kobori

Osaka Institute of Technology
1-79-1, Kitayama
Hirakata, Osaka
Japan(573-0196)
kobori@is.oit.ac.jp

ABSTRACT

We propose a method of downsizing technique for point sampled data which reduces information amount of three dimensional point cloud data. Our method generates clusters and spiral chain lists. Each chain consists of three dimensional points. After that, we adopt a predictive encoding to compress these chain lists. In addition, we show the effectiveness of our method with same experimental results by comparison with one of conventional methods. From these experiments, our method can reduce the information amount coordinate data to 31.7% of original model.

Keywords

Computer graphics, CAD, data compression, information transformation, and point cloud.

1. INTRODUCTION

Many types of 3d scanner have been developed in recent years, and we can use point cloud representation to design three dimensional shape by capturing form real shapes. This is an easy way to modeling 3d shapes from real object. So it is thought that in the near future, point cloud data will be generally distributed in CAD and computer graphics areas. Fig.1 shows an example of point cloud data. Each point has three-dimensional coordinate and normal vector.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

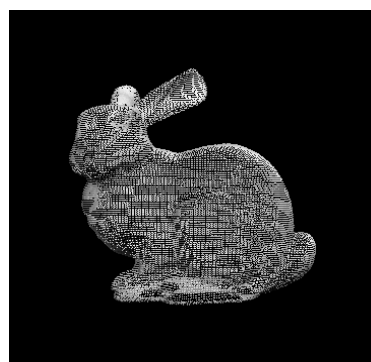


Figure 1. An example of point cloud

However, point cloud data consists of enormous number of points. As a result, it causes enlarging data size and increase of network traffic in case of transfer via networks. So it is required to compress those data by transforming them into smaller representations. Waschbüsch, et al [Was04a] proposed a method which compresses coordinates and normals progressively by using binary tree. However it is for both coordinates and normals, so

the process needs both of them and does not work separately. In addition, it is not a lossless method but a lossy one. This is the point different from our method.

In this paper, we propose an information transformation method for point cloud captured by range scanner such as 3d laser scanner, reducing its information amount. With this method, we can convert it into smaller file using conventional compression techniques such as zip and so on.

Fig.2 shows a general outline of data compression process. The process consists of two procedures, information transformation and entropy encoding. First, information transformation technique is applied to point cloud data in order to reduce information amount with redundancy elimination. After that, entropy coding is applied to compress transformed data, for example, Huffman coding, arithmetic coding and so on.

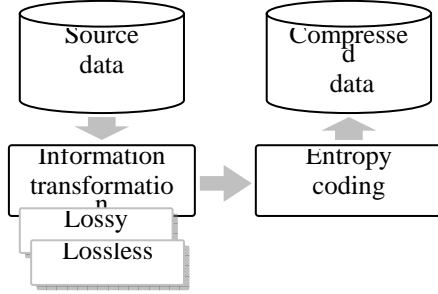


Figure 2. Outline of data compression.

2. Information transformation

First, our method generates clusters to manage data points.

In this section, we propose a new data structure. We call it Spiral Chain List(SCL). SCL consists of chain code representations listing each data point.

2.1 Point clustering

Our process generates clusters of point cloud to get more efficient result. Each cluster consists of points within a radius R_{th} defined by a user, as shown in fig.3. s_i is the center of cluster C_i . Each distance between cluster centers is longer than R_{th} . If we use bigger R_{th} , the number of clusters becomes smaller. On the other hand, if we use smaller R_{th} , the number of clusters becomes larger. The radius of clusters is the key to get better result. To get more effective result, we thought that properties of points which belong to same cluster should be similar. Those properties consist of three-dimensional coordinate, normal vector, color and so on. So we should have to

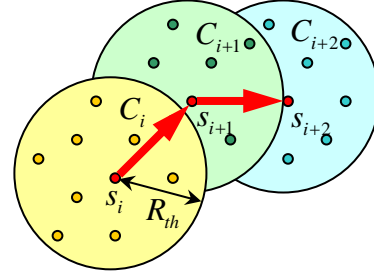


Figure 3. Clustering of point cloud and its overlapping.

choose appropriate radius R_{th} . In general, smaller radius is preferable for the point cloud which has many flat or low curvature features, and larger radius is preferable for complex one.

If cluster circles overlap each other, the number of points in each cluster gets to be smaller and the information amount of each cluster gets to be smaller too. So our method chooses centers of cluster in order to place apart from each other as long as possible. With this strategy, the process can generate large clusters efficiently and get effective result.

In the latter of the process, remaining points which do not belong to any cluster yet, are scattered and the size of cluster becomes small as process advances. However, sizes of clusters become larger as a whole result.

2.2 Spiral chain list

This process applies differential encoding to points in each cluster. Differential encoding converts properties of points into difference value. So the process generates point lists, SCL like a scroll print as shown in fig.4.

SCL connects points sequentially which begins at center of cluster s_i and glow out toward the boundary of cluster C_i . Differential values, such as differential coordinates and differential normals are calculated along SCL. In addition our method has adopted a predictive encoding. In this process, points are connected sequentially as shown in fig.5. $p_i(i = 1 \dots n)$ is a point on SCL and this process

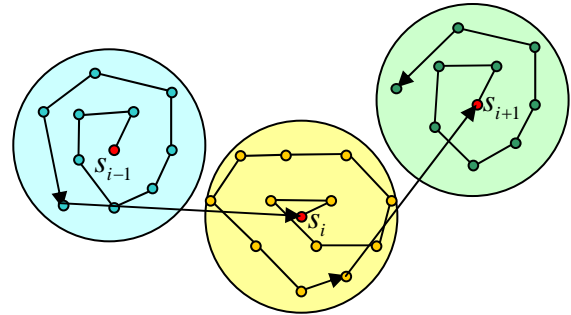


Figure 4. Spiral chain list.

calculates predictive point q . Then there is difference between next point p_{i+1} and q . So the differential vector v_{diff} is stored as a differential encoding result and in most cases, v_{diff} is smaller than v_{i+1} . With this prediction, the information amount of point cloud becomes smaller than original data.

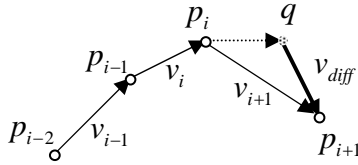


Figure 5. Prediction vector.

3. Experiments

We evaluated effectiveness of our method and measured the difference between original point cloud data and transformed data about information amounts. In this experiment, we used 2 models, Bunny and Buddha. Each model consists of 34,834 and 543,652 points with three dimensional coordinate and normal.

Experimental results are shown in table 1 and table 2. The information amount of original Bunny is 72.8KB for coordinate and 88.6KB for normal. The results of our method are 23.1KB and 86.1KB respectively. Total compression ratio is 67.7% and this is not so good. However, the compression ratio for coordinate is 31.7%. The results for Buddha are 195.4KB, 852.7KB and 1046.1KB respectively. Total compression ratio is 44.1%, and Coordinate compression ratio is 19.7%.

Fig.6 and 7 show frequencies of original and transformed Bunny using 8-bit quantization. Fig 8 shows the path of spiral chain list generated by proposed method and its close up image is shown in fig.9.

The results of Buddha are shown in fig 10, 11 and 12.

From this experiment, our method is effective just for coordinate to reduce information amount of point cloud. It is thought that in our method, the predictive encoding is applied only to coordinate, so compression ratio for normal is not effective. We think this may be improved by adopting prediction to normal also.

Table 1. Information amounts for Bunny.

	Original	Transformed	Ratio
Coordinate	72.8KB	23.1KB	31.7%
Normal	88.6KB	86.1KB	97.2%
Total	161.4KB	109.2KB	67.7%

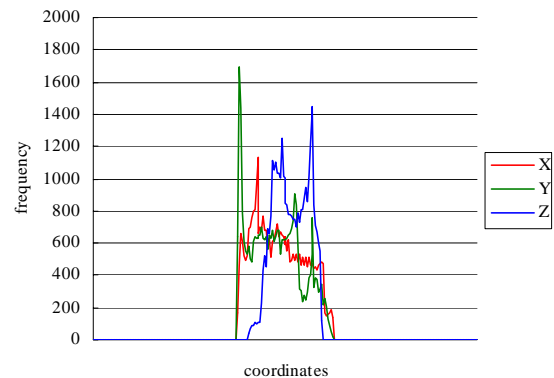
Table 2. Information amounts for Buddha.

	Original	Transformed	Ratio
Coordinate	989.7KB	195.4KB	19.7%
Normal	1388.2KB	852.7KB	61.4%
Total	2377.8KB	1048.1KB	44.1%

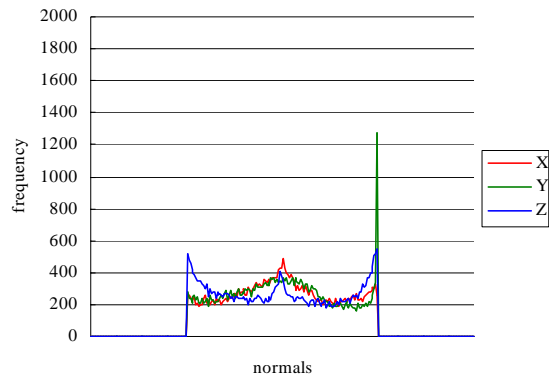
4. Conclusions

From this experiment, our method is effective to reduce information amount of point cloud and provide more compact representation for point cloud. It is thought that our method is effective for coordinate to reduce data size and reduction of network traffic when point cloud data is transferred via network.

Now we're trying to implement one of compression techniques to reduce information amount of point cloud. We will show the results of comparison our method and conventional one.

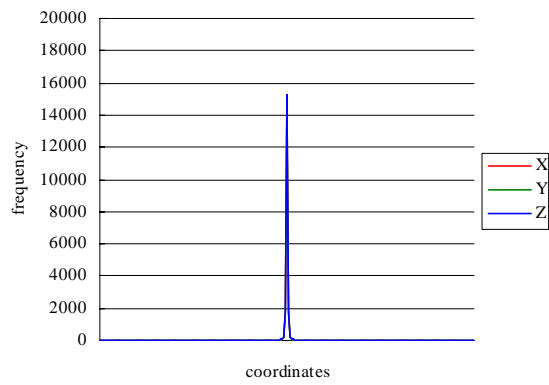


(a) Coordinates

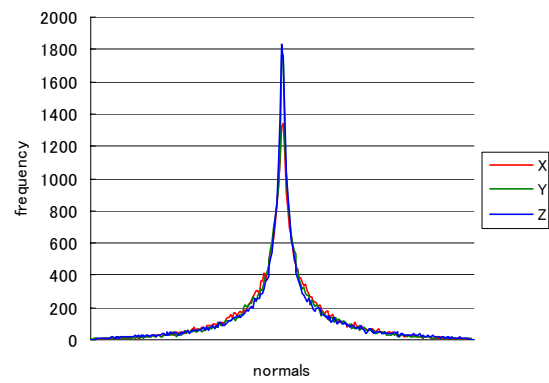


(b) Normals

Figure 6. Histograms of original Bunny.



(a) Coordinates



(b) Normals

Figure 7. Histograms of transformed Bunny.

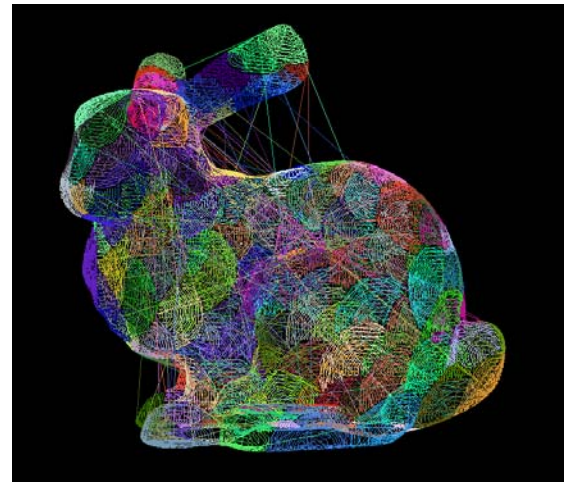


Figure 8. Path of spiral chain list for Bunny.

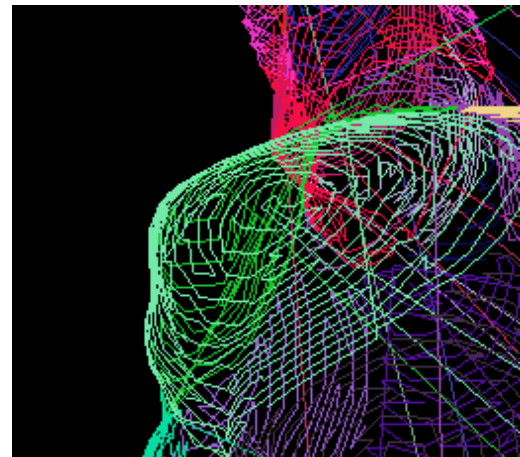
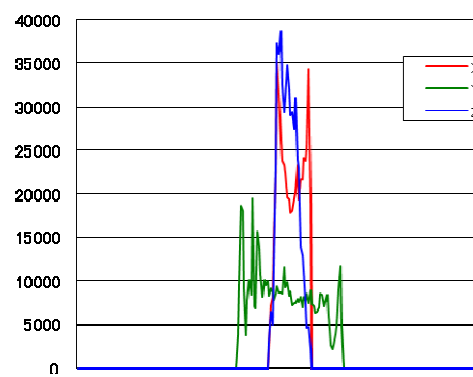
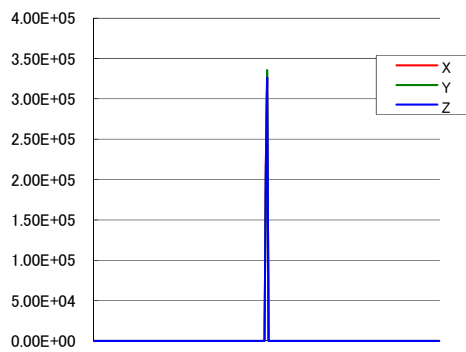
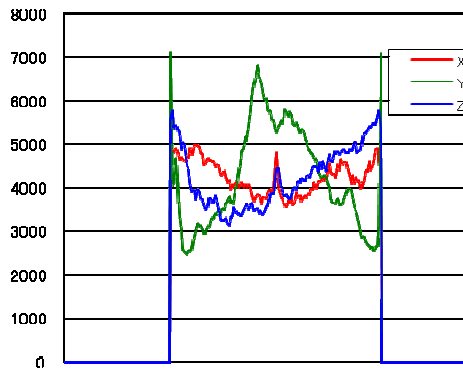


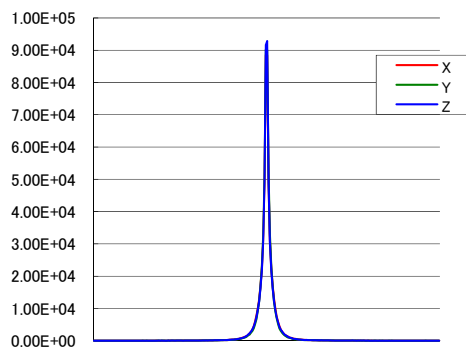
Figure 9. Detail of spiral chain list.



(a) Coordinates



(a) Coordinates



(b) Normals

Figure 11. Histograms of transformed Buddha.

(b) Normals

Figure 10. Histograms of original Buddha.



Figure 12. Path of chain list for Buddha.

5. REFERENCES

[Was04a] Waschbüsch, M.. Progressive Compression of Point-Sampled Models. Proceedings of the Eurographics Symposium on Point-Based Graphics 2004, pp. 95-102.

Improving the Responsiveness in Multiplatform Collaborative Environments

Luiz Gonzaga da Silveira Jr
UNISINOS, V3D Studios, Brazil
lgonzaga@unisinos.br

Wu Shin-Ting
UNICAMP, Brazil
ting@dca.fee.unicamp.br

ABSTRACT

This paper addresses the design issues that may improve the responsiveness of a multi-platform collaborative modeling system, for which robustness and awareness are necessary requirements. The key points of our proposal are, whenever possible, (1) to reduce as much as possible the granularity of the transmission data over network; (2) to simplify as much as possible the functional feedbacks for fast screen update; and (3) to avoid as much as possible the network accesses for synchronization. On the basis of these hypotheses, we explore the features of a hybrid groupware architecture and show the feasibility of our proposal. Several latencies are measured to validate our assumptions.

Keywords: Collaborative Systems, Groupware, Geometric Modeling, Geometric Robustness, Awareness.

1 INTRODUCTION

The typical scenario for collaborative modeling is a shared workspace, where a dispersed group of users (end-users) work together for creating and modifying an application-dependent 3D-model over an internet network. Concerning with the underlying system architecture one may distinguish three approaches: centralized, replicated [2], and hybrid one [9].

In the centralized architecture only one instance of the shared application runs in a central server, while end-user workspaces display the same scene from the central server and managing the input events. It makes the 3D-model concurrency control simpler to be implemented, but the interactivity might be compromised. Beside, this approach may generate substantial overload both in the central server and network due to the continuous traffic and processing.

In the replicated architecture, one instance of the shared application runs locally on each end user's workspace. The system's response time may be enhanced, once the network traffic is relatively lighter. The benefits of a replicated architecture must, however, be balanced against the homogeneous numerical computation offered by the centralized one, when we migrate to a heterogeneous computing environment. Under heterogeneous platform, we understand internetwork of computers equipped with distinct hardwares (CPU, display technologies, memory and mainly GPUs), under different operating systems, com-

piled implementations and capabilities for float-point computing.

To achieve consistency across the heterogeneous machines, without disregarding their individual performance, we proposed in [3, 9] a hybrid architecture for collaborative applications, as a tradeoff solution for keeping the consistency of 3D-model (geometric robustness) and keeping the system usability. The hybrid architecture results from the combination of the both centralized and replicated architectures. The basic idea consists in separating application-dependent model from graphics functionalities.

Besides the robustness, the separation of the geometric and the graphical model makes the rendering mode in each participating machine tailorable to the local computing power. Though, the participants may not only be working in different parts of the space with distinct viewpoints, but also calibrate the rendering parameters to the acceptable interactivity level.

In this paper we consider the aspect that helps ensure usability of any interactive system: the responsiveness. We will show that in the hybrid architecture we may control the granularity and the frequency of the transmitted information without sacrificing interactivity and robustness. Moreover, since the rendering mode may be tuned in the local workspace to fit the hardware capabilities. Personalized rendering may be set to compensate the latencies of different network transmission rates and the distinct processing performance of low or high end GPUs used together. To validate our proposal, we have integrated our solutions in the multiplatform collaborative geometric modeler called CoMo (Collaborative Geometric Modeler) [9]. Several measures of latency have been performed and compared with the range of acceptable values proposed by Nielsen [7].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WSCG'2010, February 1 – February 4, 2010
Plzen, Czech Republic

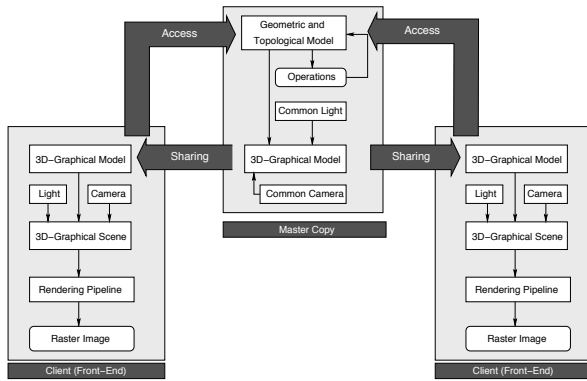


Figure 1: The conceptual model of a collaborative modeling system.

2 DESIGN ISSUES

In [9], we presented a hybrid architecture for collaborative 3D modeling systems, where two equivalent data are kept in the *geometric modeling kernel*: a geometric and a graphical data (Figure 1).

The geometric data is common to all end-users (interactive applications), whereas the 3D graphics data are replicable for visualization and manipulation in each end user workspace (*interactive application*). In this approach, we may take advantages of the well-known robust geometric algorithms [5], designed for monolithic modeling applications, to enhance the robustness of the entire heterogeneous platform.

To provide a more versatile way to refer to an object residing in the *geometric modeling kernel*, a proxy [1] is designed to make the user workspace communicate with a representative of the geometric model, rather than with the geometric model itself. We reused rendering and interaction functionalities provided by the Manipulation Toolkit *MTK* [4], running on top of OpenGL [6]. This is because that a main differential of *MTK* with respect to the other known graphical toolkits is its loosely decoupling of the application and the graphical models, although it provides efficient direct manipulation mechanisms via 3D-metaphors.

Several users may interact with the shared 3D model simultaneously. To avoid/solve resource contention of potential conflicts that may arise from simultaneous accesses of a shared application by various end users and to support group awareness, it is also devised in our architecture *floor control mechanisms* residing in a *group manager server*. The adopted infrastructure for objects communications over a network, independent of specific platform and techniques used to implement these objects, is based on the the Common Object Request Broker Architecture (CORBA) [8].

One drawback of the hybrid architecture, is that the group awareness may be drastically reduced and the system's usability may be deteriorated. As a solution, we proposed to integrate two awareness sub-windows

in the user interface of each interactive application (on the left side of each application interface in Figure 2), in addition to the conventional drawing area (*scene view*) where users can interact with the 3D model through the 3D-metaphors (on the right side of each application interface in Figure 2). The awareness windows are responsible for conveying the global view of the ongoing activities: one is a listbox that contains the participant names (on the bottom left) and the other is the second drawing area (*global view* on the top left) where a simplified version of the global overview of the 3D shared workspace is presented.

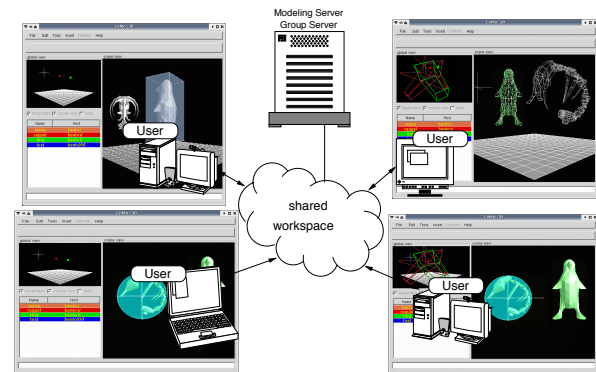


Figure 2: A Sample of The Modeller User Interface

For 3D graphics interactive systems, some of usability metrics are functions of system's latency. Latency is related with the update speed of an image in response to a user action. It plays an important role in the fluidity of end user interactions with their applications. The latency must be the lowest as possible. As computer cannot provide fairly immediate response, three important latency limits have been identified regarding the reaction and behavior of end users [7]: 0.1s - the system is reacting instantaneously, no feedback is required; 1.0s - limit for the user's flow of thought to keep uninterrupted, despite the noticeable delay; 10s - the limit for keeping the user's attention focused on the dialogue, visual feedback is required.

3 THE SYSTEM RESPONSIVENESS

The emphasis on the design of an end-user workspace application for a hybrid architecture based system is its tailorability to each local computing power and the accessibility to all geometric functions provided by a geometric modeling server and the knowledge of the actions of the other users that share the same application model.

For making each instance of the end user workspace an interactive application, we have proposed useful awareness feedbacks, four problems must be solved:

1. System latency;
2. Event handling;

3. Remote pointing and viewing volumes;
4. Graphic states synchronization;

3.1 System latency

One of the challenging issues that we must circumvent is to devise a technique for interacting with a remote object through its corresponding graphical object in the user workspace, whose latency can be expressed as

$$t_{latency} = t_{t1} + t_m + t_{t2} + t_r, \quad (1)$$

where t_{t1} , t_m , t_{t2} , and t_r , are, respectively, the transmission time of a user's request, the processing time in the central server, the transmission time of the server's response, and the rendering time on each participating computer.

Eq. 1 suggests us that when an application is separated from the user interface, we have several ways to improve its latency: (1) we may locally adjust the rendering parameters (t_r) in order to counterbalance the delays in a network (t_{t1} and t_{t2}); (2) we may invest in the processing power of the central server (t_m); or (3) we may invest in a higher bandwidth network. Whatever is the solution, a hybrid architecture supports it.

Our network-independent solution for optimizing the interaction performance is a graphical object–dragger loosely coupling interaction paradigm that is supported by *MTK*. Under a dragger, we understand an object that has a pictorial representation and can map the 2D inputs from the pointing devices to motions in three dimensions. With the Mediator design pattern [1], we define the class of objects, called Manipulator, to control the interaction between each pair dragger–graphical object. In the graphical object–dragger interaction model, two interaction loops may be distinguished: user–dragger–geometrical model–graphical model–user and user–dragger–user (Figure 3).

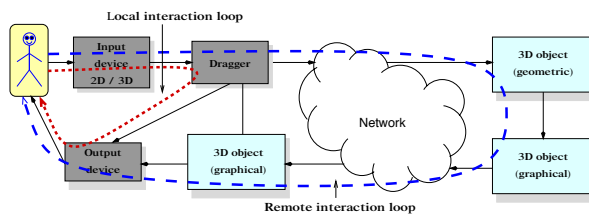


Figure 3: Interaction loops.

It is important to remark that the time response of the two interaction loops are different. Whereas the response time of the first loop is given by Eq. 1, the latency of the second loop may be expressed by

$$t_{dragger_latency} = t_{lp} + t_r, \quad (2)$$

where t_r and t_{lp} are, respectively the local processing and rendering time.

Observing in Figure 3, by a sequence of events view-point, a manipulator generates a unit of information that

is a semantically valid transformation, which is applied on the the dragger for local visual feedback. In parallel, the sequence of transformations is concatenated into a unique transformation for updating a 3D geometric object in the geometric server. This approach relieves the communication traffic.

3.2 Event Handling

The cost for loosely decoupling the geometric and graphical model in the context of interactions is the increase in the complexity of event handling. In addition to the user input events that can be handled by any interaction techniques toolkit, there are events from the communication channel that may also affect the context of the *scene view* sub-window, as illustrated in Figure 3.

An algorithm is necessary to extend the standard dispatching code for selecting the correct window for each of these events. A solution is to use an interaction that allows some portions of the event dispatching code to be modified by application programmers, provided by all GUI SDKs. On top of the window system, a Chain of Responsibility [1] defines an object that decouples the sender of events from the windows whose handlers an event should be forwarded.

3.3 Remote Pointing and Viewing Volume

Whenever a user interacts with the pointing device, the sequence of actions is collected and mapped into a meaningful unit of information. This unit of information is broadcasted to all the rest of participating machines for updating the state of the replicated manipulator. It guarantees the location awareness. Moreover, if the viewing parameters at each user workspace application are modified, they must also be multicasted to all the participating machines for redisplaying the context of the global view window and maintaining the perspective awareness. We propose an Observer design pattern [1] to define the object that performs this passive replication: the original manipulator/view parameters is the subject that all their replica (observer) must keep on observing. This approach keeps all client applications consistently updated.

It is worth observing that the latency of the manipulators is the most critical one, since the user interacts continually with them and the obtained units of information must be constantly transmitted over the network. This latency depends on the transmission time of the replicated data t_{t1} and the rendering time t_r in each participating machine (Figure 4):

$$t_{rep_latency} = t_{t1} + t_r. \quad (3)$$

To make this latency as lower as possible, we suggest to use the wireframe rendering mode in the global view window and to adopt the simplest graphical representation to the replicas. The pictorial representations of

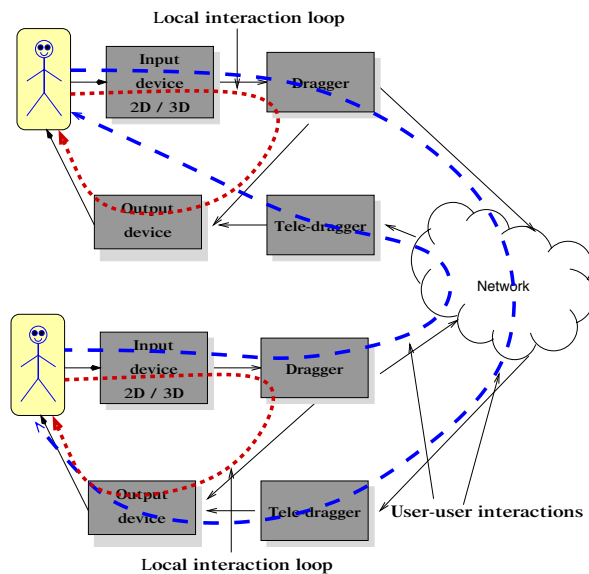


Figure 4: User-user interactions.

the pointing devices and the volume view that we chose are, respectively, colored graphical representations of the manipulators and colored wireframe boxes. Furthermore, we allocate an event channel only for replications of the manipulators and view volumes.

3.4 Graphics Attributes Synchronization

For providing appropriate visual feedbacks, positions of a pointing device that a user manages should be shown in the *scene view* sub-window. At the same time, for the sake of group awareness, these positions must be sent to all instances of the user workspace application, inclusive the instance that generates the event, in order to update the content of every *global view* sub-window. This means that both sub-windows are selected and their corresponding event handlers are invoked for, concurrently, redisplaying. These handlers need to access correct graphics states for correctly re-rendering the graphics objects. Otherwise, incorrect drawings may be generated.

The solution that we propose for synchronizing the graphics states with the selected windows is to explicitly issue the command that make the graphics states of any focused window as the current states. In this way, the handler can always draw the objects with the expected attributes in each window.

4 EXPERIMENTS AND CONCLUDING REMARKS

To validate our proposal, we implemented a new version of CoMo with extended functionalities, installed in different machines (Sparc with Elite 3D, and PCs with FX6600 and GeForce 8600GT), and measured latency parameters over 1.0Mb/s, 10.0 Mb/s and 1.0Gb/s

LANs. We collected some measure about latencies parameters proposed in this paper.

The dragger latency lied in the range 0.1–1.0s for all machines. While, the results collected for object manipulation shows the transmission rate did not affect the object latency. It's because the size of the geometric data was much smaller than the network capacity. It is one of the advantages of replicated and hybrid architectures over a centralized architecture. The tele-dragger latency has been the most critical one, since the user interacts continually with it. Therefore, we shows that the latency of the tele-draggers is dominantly dependent on the network transmission rate.

Based on experiment, our proposed solutions is close, but does not satisfy completely yet, the recommended interactivity metrics. We have work in thee system reimplementaion, considering lightweight communication protocols.

The main contribution of this work is to demonstrate that it is feasible to design an interactive and usable system in a heterogeneous multi-platform environment, where the shared data consistency must be ensured. Besides, rendering parameters and display can be adapted accordingly with local system resource to assure individual performance. We believe that a hybrid architecture may becomes a good alternative for any multi-platform application whose the most important requirements are the robustness and the adaptability to local computational resources.

REFERENCES

- [1] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Desin Patterns – Elements of Reusable Object-Oriented Software*. Addison Wesley, 1995.
- [2] S. Greenberg, S. Hayne, and R. Rada. *Designing Groupware for Real-Time Drawing*. McGraw Hill, 1995.
- [3] L.G. Silveira Jr and S.-T. Wu. An object-oriented groupware framework for developing collaborative 3d-modelers. In Thierry Priol e Jamie Painter, editor, *Third Eurographics Workshop on Parallel Graphics & Visualisation*, pages 103–114, Girona, ESP, Sept. 2000.
- [4] M. de G. Malheiros, F. N. Fernandes, and S. T. Wu. Mtk: A direct 3d manipulation toolkit. In *SCCG'98 Proceedings*, pages 81–88, Brastilava, april 1998.
- [5] D. Michelucci. An introduction to the robustness issue. In *Swiss Conference of CAD/CAM*, pages 214–221, Neuchâtel, Switzerland, Feb. 1999.
- [6] J. Neider, T. Davis, and M. Woo. *OpenGL - Programming Guide - Release 1*. Addison Wesley Co., 1993.
- [7] Jakob Nielsen. *Usability Engineering*. Morgan Kaufmann Publishers, 1994.
- [8] Object Management Group (OMG). The common object request broker: Architecture and specification, Dez 2001. Version 2.6.
- [9] L.G. Silveira Jr and S.-T. Wu. Towards consistency in a heterogeneous collaborative geometric modeling environmen. In *Proceedings of SIACG 2002*, pages 139–148, Guimarães, Portugal, 1–5 July 2002.